An Effective Reliability Efficient Algorithm for Enhancing the Overall Performance of Distributed Computing System

Pankaj Saxena
Teerthanker Mahaveer University, Moradabad (U.P), India

Kapil Govil, Ph.D
Teerthanker Mahaveer University, Moradabad (U.P), India

ABSTRACT
Distributed computing refers to the use of distributed systems to solve computational problems. A distributed computing system consists of multiple computers that communicate through a computer network. The computers that are in a distributed computing system can be physically close together and connected by a local network, or they can be geographically distant and connected by a wide area network. Distributed computing systems offer the benefits like scalability and redundancy. A task is any single module to be processed. If the number of tasks are more then the number of processors and every processor process the task in a particular time period for processing any particular task then we have to allocate each task to the single processor in such a way that the task should be completed in a optimal reliability manner and also there should not be overloading of task to any single processor. The number of processors and number of tasks are static in nature. The number of processors is denoted by n and the number of tasks is denoted by m. In general for all real world problem the number of tasks are greater then the number of processors i.e. m>n. The requirement is to complete all the tasks by allocating the task so that the results for reliability should be optimal in nature to increase the overall performance of distributed computing system.

Keywords
Distributed Computing System, Task, Processor; Task allocation.

1. INTRODUCTION
In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers which communicate with each other. Distributed computing system [25, 29] referred to computer networks where individual computers are physically distributed. In this paper the task is allocated statically [6, 15] to the processor. Task allocation [1, 8, 11] refers to assign the task on the available processor for increasing the performance of distributed computing system [5, 24]. Task allocation [13, 17, 18] is in the way to enhance the overall processing reliability [3, 7, 10]. Also to improve the performance [19] of distributed network [2] it is required to maximize the processing reliability [12, 20, 21]. An important aspect of distributed computing is to provide a user a non distributed view of a distributed system. In the present paper we are giving an algorithm [14, 27, 28] to be implemented in a distributed system and where the numbers of tasks [4, 9, 16] are greater then the number of processors.

2. OBJECTIVE
The objective of the present research paper is to give an efficient and reliable algorithm to allocate all the available tasks on processors statically in a distributed computing system to enhance the results for processing reliability. In distributed computing, there is a network of many processors where each processor accomplishes the task to get the optimal result more quickly as well as more efficiently. Here we have taken an example of distributed computing system where the number of processors is lesser in comparison to the number of tasks. In this research paper there is a set of tasks which are to be get processed by some processors. The objective also includes, calculating the time complexity of the present algorithm as it is a major factor to show the performance of the algorithm. The objective part of present paper also contains the comparison between results with some other recent algorithm for proving the betterment of the present algorithm.

3. NOTATIONS

n	:	Number of processors
m	:	Number of tasks
P	:	Processor
T	:	Task
PRM	:	Processor Reliability Matrix
MPRM	:	Modified Processor Reliability Matrix
APRM	:	Arranged Processor Reliability Matrix

4. TECHNIQUE
In order to evaluate the overall optimal processing reliability [22, 23, 26] of a distributed Computing System [2] we have chosen the problem where a set P= $\{p_1, p_2, p_3 \ldots p_n\}$ of 'n' processors and a set T= $\{t_1, t_2, t_3 \ldots t_m\}$ of 'm' tasks, where m>n. The processing time of each and every task to each and every processor is known and it is mentioned in the processing reliability matrix namely PRM (,) of order n*m. On making the addition of each column of PRM (,) we find MPRM (,) and after arranging MPRM (,) it in ascending order of their sum of column we find Arranged Processing Time Matrix namely APRM (,).

5. ALGORITHM
Step 1: Read the number of tasks in m
Step 2: Read the number of processor in n
Step 3: Read the Processing reliability Matrix PRM (,) of order n*m
Step 4: Compute multiplication of each column of PRM (,) and store it into Modified Processor Reliability Matrix MPRM (,)
Step 5: Arrange the MPRM (,) in ascending order of their column multiplication value and store it into Arranged Processor Reliability Matrix APRM (,)
Step 6: While (all tasks! = allocated)
Step 7: {
Select and allocate particular task from APRM (,) processor which has the maximum Processing reliability.

}

Step 8: Compute the processor wise overall processing reliability.

Step 9: Display the result.

6. IMPLEMENTATION

In the present research paper we have taken total number of task 10 which is $t_1,t_2,t_3,t_4,t_5,t_6,t_7,t_8,t_9$ and t_{10}, and total number of processors 4,which are p_1,p_2,p_3 and p_4.diagramatically it can be shown below in figure1-

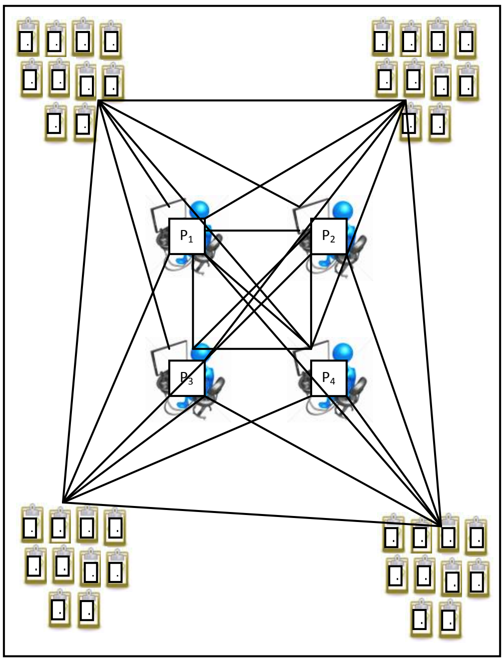

Fig 1: Task Allocation Problem

The processing reliability (r) of each task on various processors are known and mentioned in the processing reliability matrix namely PRM (,) of order 4 * 10-

PRM(,) =

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
p_1	0.999567	0.999712	0.998877	0.999334	0.998543	0.999865	0.998855	0.999812	0.999871	0.998124
p_2	0.999723	0.998511	0.999833	0.998865	0.998877	0.999888	0.997755	0.987652	0.999712	0.999432
p_3	0.998532	0.999777	0.991122	0.998765	0.997654	0.999777	0.993322	0.998765	0.999431	0.999654
p_4	0.998762	0.999877	0.999222	0.998765	0.998887	0.999666	0.991223	0.987345	0.998745	0.999763

On making the multiplication of each column of PRM (,) we find Modified Processing Reliability Matrix namely MPRM (,) of order 4 *10-

MPRM(,) =

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
p_1	0.999567	0.999712	0.998877	0.999334	0.998543	0.999865	0.998855	0.999812	0.999871	0.998124
p_2	0.999723	0.998511	0.999833	0.998865	0.998877	0.999888	0.997755	0.987652	0.999712	0.999432
p_3	0.998532	0.999777	0.991122	0.998765	0.997654	0.999777	0.993322	0.998765	0.999431	0.999654
p_4	0.998762	0.999877	0.999222	0.998765	0.998887	0.999666	0.991223	0.987345	0.998745	0.999763
$\prod$	0.996587	0.997878	0.989073	0.995735	0.993974	0.999196	0.981268	0.973765	0.997760	0.996975

On arranging MPRM (,) in ascending order of their column multiplication value, we find Arranged Processor Reliability Matrix APRM (,) of order n*m-

APRM(,) =

	t_6	t_2	t_9	t_{10}	t_1	t_4	t_5	t_3	t_7	t_8
p_1	0.999865	0.999712	0.999871	0.998124	0.999567	0.999334	0.998543	0.998877	0.998855	0.999812
p_2	0.999888	0.998511	0.999712	0.999432	0.999723	0.998865	0.998877	0.999833	0.997755	0.987652
p_3	0.999777	0.999777	0.999431	0.999654	0.998532	0.998765	0.997654	0.991122	0.993322	0.998765
p_4	0.999666	0.999877	0.998745	0.999763	0.998762	0.998765	0.998887	0.999222	0.991223	0.987345
$\sum$	0.999196	0.997878	0.997760	0.996975	0.996587	0.995735	0.993974	0.989073	0.981268	0.973765

Now we select first 4 tasks (t_6,t_2,t_9,t_{10}) from APRM(,), as we have 4 processors (p_1,p_2,p_3,p_4). On selecting first 4 tasks which have the maximum processing reliability at any specified processor, we find the allocations which are mentioned in Table 1.

Table1 : Task Allocation

Processor	Allocated task	Proessing Reliablity
p_1	t_9	0.999871
p_2	t_6	0.999888
p_3	t_{10}	0.999654
p_4	t_2	0.999877

Again, we select next four tasks (t_1, t_4, t_5, t_3) from APRM (,).Again on selecting 4 tasks which have the maximum processing reliability at any specified processor, we find the allocations which are mentioned in Table 2.

Table 2 : Task Allocation

Processor	Allocated task	Proessing Reliablity
p_1	t_1	0.999567
p_2	t_3	0.999833
p_3	t_4	0.998765
p_4	t_5	0.998887

Here, we have only two tasks (t_7, t_8) more. On repeating the similar concept we select tasks which have the maximum processing reliability at specified processor, we find the allocation which are mentioned in Table 3.

Table 3 : Task Allocation

Processor	Allocated task	Proessing Reliablity
p_1	t_8	0.999812
p_2	t_7	0.997755
p_3	…	…
p_4	…	…

Now the overall allocations on the various processors are mentioned in Table 4.

Table 4 : Task allocation

Processor	Task allocation	Processing reliablity
p_1	$t_9 * t_1 * t_8$	0.999250
p_2	$t_6 * t_3 * t_7$	0.997476
p_3	$t_{10} * t_4$	0.998419
p_4	$t_2 * t_5$	0.998764
Overall Processing Reliablity	:	0.993922

The graphical representation is described by figure 2 which is given below-

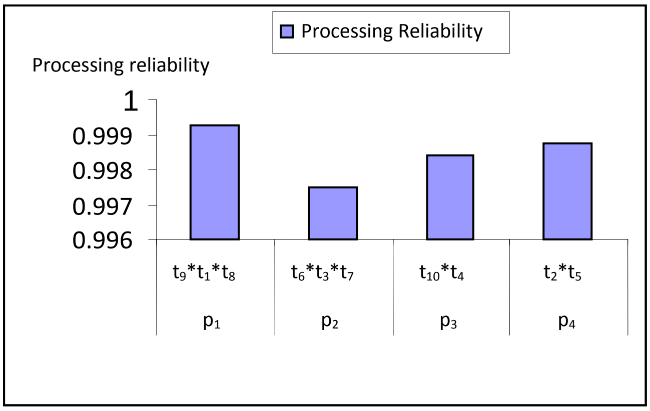

Fig 2: Processing Reliability based Task Allocation

7. CONCLUSION

In the present paper the problem is discussed to allocate the different tasks to different processors. The number of tasks is more then the number of processors. The tasks are allocated to the processors to get the optimum utilization of processors in terms of enhanced processing reliability for each task. The concept in the present paper is presented in algorithmic form and it is tested for many inputs to check the accuracy of result. The model mentioned in this paper is based on the consideration of processing reliability of the tasks to various processors. It is the common requirement for any assignment problem that the tasks have to be processed with maximum reliability. Here, performance is measured in terms of processing reliability of the task that has been processed by the processors of the network and also these tasks have been processed optimally. The optimal result which is also mentioned in the implementation section of the paper, is as given below-

Table 5 : Results

Processor	Task	Optimal Results
p_1	$t_9 * t_1 * t_8$	
p_2	$t_6 * t_3 * t_7$	0.993922
p_3	$t_{10} * t_4$	
p_4	$t_2 * t_5$	

As we know that, the analysis of an algorithm is mainly focuses on time complexity. Time complexity is a function of input size 'n'. It is referred to as the amount of time required by an algorithm to run to completion. The time complexity of the above mentioned algorithm is O (mn). By taking several input examples, the above algorithm returns following results,

Table 6 : Results

No.of Processors(n)	No.of Tasks(m)	Optimal Results
4	5	20
4	6	24
4	7	28
4	8	32
5	6	30
5	7	35
5	8	40
5	9	45
6	7	42
6	8	48
6	9	54
6	10	60

The graphical representation of the above results are shown by figure 3, 4 and 5 as mentioned below-

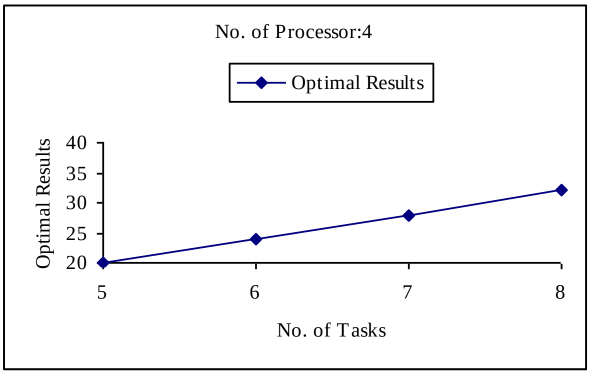

Fig 3: Graphical Representation

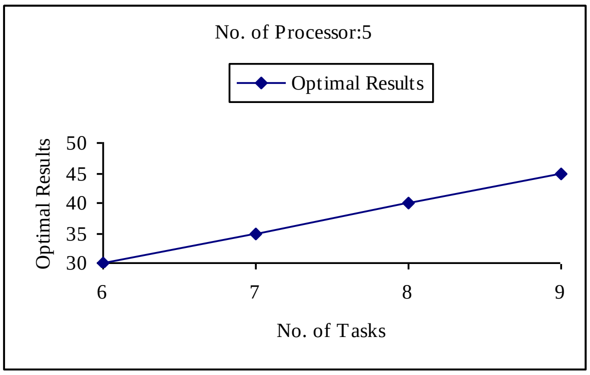

Fig 4: Graphical Representation

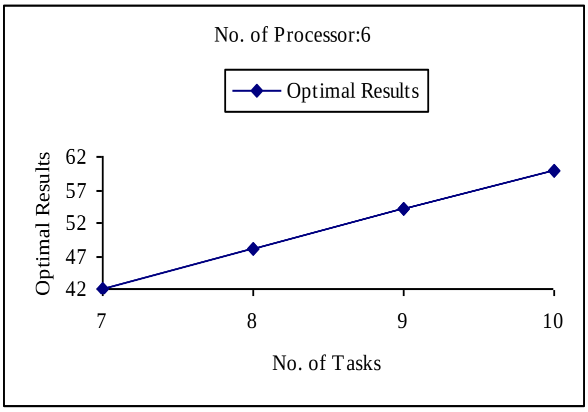

Fig 5: Graphical Representation

The performance of the algorithm is compared with the algorithm suggested by Liu et.al (14). Following table shows the time complexity comparison between algorithm (14) and present algorithm-

Table 7 : Time Complexity

Processors(n)	Tasks(m)	Time Complexity of algorithm(14) $O(n^3 \log n)$	Time Complexity of Present algorithm O(mn)
4	5	38	20
4	6	38	24
4	7	38	28
4	8	38	32
5	6	87	30
5	7	87	35
5	8	87	40
5	9	87	45
6	7	168	42
6	8	168	48
6	9	168	54
6	10	168	60

From the above table it is clear that present algorithm is much better for optimal allocation of tasks that upgrade the performance of distributed computing system. Following graphs in figure 6, 7, and 8 shows the pictorial representation between algorithm (14) and present algorithm-

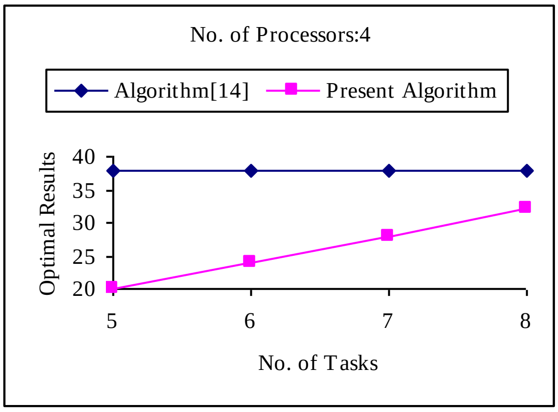

Figure 6: Graphical Representation

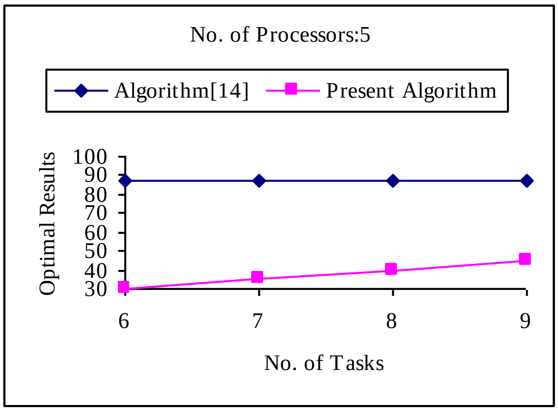

Figure 7: Graphical Representation

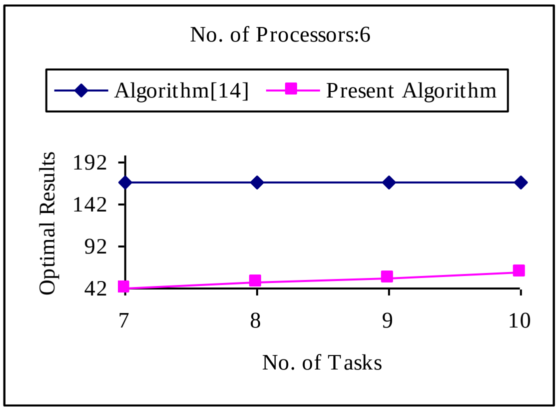

Figure 8: Graphical Representation

8. REFERENCES

[1] Anil Kumar tripathi,deo prakash vidyarthi,A.N.mantri, "a genetic task allocation algorithm for distributed computing systems incorporating problem specific knowledge", international journal of high speed computing,vol:8, issue:4, pp:363-370.

[2] Asaduzzaman Shah, Maheswaran Muthucumaru, "Decentralized management of bi-modal network resources in a distributed stream processing platform", Journal of Parallel and Distributed Computing, vol: 71, issue 6, pp: 774-787, 2011.

[3] Anurag raii, vikram kapoor, Reliable Clustering Model for Enhancing Processors Throughput in Distributed

Computing System, international journal of computer applications, vol:38, issue:8, pp:0975-8887, 2012.

[4] Ashish chandak, bibhodatta sahoo, ashok kumar turuk, "Task Scheduling Heuristic in Grid Computing, International Journal of Computer Applications and Technology" ,vol:1, issue:2, pp:49-52, 2012.

[5] Bibhudatta Sahoo, Dilip Kumar, Sanjay Kumar Jena, "Analysing the Impact of Heterogeneity with Greedy Resource Allocation Algorithms for Dynamic Load Balancing in Heterogeneous Distributed Computing System", International Journal of Computer Applications, vol:62, issue:19, pp:25-34, 2013.

[6] Dr. Kapil Govil, and Dr. Avanish Kumar, "A Modified and Efficient Algorithm for Static Task Assignment in Distributed Processing Environment," International Journal of Computer Applications, vol.23, pp.0975 – 8887, June 2011.

[7] Dr. Kapil Govil, "Processing Reliability based a Clever Task Allocation Algorithm to Enhance the Performance of Distributed Computing Environment," Int. J. Advanced Networking and Applications, vol. 03, pp.1025-1030, 2011.

[8] G.sagar, anil K, sarj E, "Task allocation model for distributed systems", International Journal of Systems Science, vol: 22, issue: 9, pp: 1671-1678, 1991.

[9] H. J. Siegel, S. Ali, Techniques for mapping tasks to nodes in heterogeneous computing Systems, Journal of Systems Architecture,vol:46, issue:8, pp:627-639, 2000.

[10] Hsieh, Chung-Chi, Hsieh, Yi-Che, "Reliability and cost optimization in distributed computing systems", journal of Computers & Operations Research, vol: 30, issue: 8, pp: 1103-1119, 2003.

[11] Iqbinderpal Singh, Maninder Singh, Cloud Service Allocation based on Service Effectiveness and User Requirement, international journal of emerging trends and technology in computer science, vol:2, issue:1, pp:190-193, 2013.

[12] I.stephie Rachel, Joshua Samuel raj, v.vasudevan, A Reliable Schedule with Budget Constraints in Grid Computing, international journal of computer applications, vol:64, issue:3, pp:0975-8887, 2013.

[13] Kołodziej Joanna, Xhafa Fatos, "Modern approaches to modeling user requirements on resource and task allocation in hierarchical computational grids", International Journal of Applied Mathematics and Computer Science, vol: 21, issue: 2, pp: 243–257, 2007.

[14] Lantao Liu, Dylan Shell, A New Distributable Assignment Algorithm, 2013.

[15] Marwa shouman, gamal attiya, ibrahim Z.morsi, Static Workload Distribution of Parallel Applications in Heterogeneous Distributed Computing Systems with Memory and Communication Capacity Constraints, international journal of computer applications, vol:34, issue:6, pp:0975-8887, 2011.

[16] Mohammad I. Daouda, Nawwaf Kharma, A hybrid heuristic–genetic algorithm for task scheduling in heterogeneous processor networks, Journal of Parallel and Distributed Computing, vol:71, pp:1518-1531, 2011.

[17] Mostapha zbakh, said el hajji, "Task allocation problem as a non cooperative game", Journal of Theoretical and Applied Information Technology, vol: 16, issue: 2, pp: 110-115, 2010.

[18] Manisha Sharma, Harendra Kumar, Deepak Garg, "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, issue: 1, pp: 50-55, 2012.

[19] Pankaj Saxena, Kapil Govil, An Optimized Algorithm for Enhancement of Performance of Distributed Computing System, International Journal of Computer Applications, Volume 64, issue:.2, PP:37-42, 2013.

[20] Pradeep Kumar Yadav, M.P. Singh, Kuldeep Sharma, "Task Allocation Model for Reliability and Cost optimization in Distributed Computing System", International Journal of modeling, simulation and scientific computations, vol: 2, issue:2, pp. 1-19,2011.

[21] Peng-Yeng Yin, Shiuh-Sheng Yu, Pei-Pei Wang, Yi-Te Wang , "Task allocation for maximizing reliability of a distributed system using hybrid particle swarm optimization" , Journal of Systems and Software, vol:80, issue: 5, pp: 724-735, 2007.

[22] Qin-Ma Kang, Hong He, Hui-Min Song, Rong Deng, "Task allocation for maximizing reliability of distributed computing systems using honeybee mating optimization", Journal of Systems and Software, vol: 83, issue: 11, , pp: 2165–2174, 2010.

[23] Sa meng, yanping xiang, huijuan fan, shengji yu, maximal profit service task partition and allocation in computer grid considering service reliability and security, journal of theoretical and applied information technology,vol:50, issue:1, pp:260-268.

[24] S. F. El-Zoghdy, M. Nofal, M. A. Shohla, A. El-sawy, An Efficient Algorithm for Resource Allocation in Parallel and Distributed Computing Systems, International Journal of Advanced Computer Science and Applications, vol: 4, issue: 2, pp:251-259, 2013.

[25] Sol M. Shatz, Jia-Ping Wang, Masanori Goto, "Task Allocation for Maximizing Reliability of Distributed Computer Systems", IEEE Transactions on Computers, Vol: 41 issue :9, pp: 1156-1168 , 1992.

[26] Santhanam Srinivasan, Niraj K. Jha, "Safety and Reliability Driven Task Allocation in Distributed Systems", IEEE Transactions on Parallel and Distributed Systems, vol: 10 issue: 3, pp: 238-251, 1999.

[27] Tejinder Sharma, Vijay Kumar Banga, Efficient and Enhanced Algorithm in Cloud Computing, International Journal of Soft Computing and Engineering, vol: 3, issue: 1, pp:385-390, 2013.

[28] Vivekananth.P, Eigen Trust Algorithm for Resource Selection in Grid Computing, International Journal of Computer Applications, vol: 16, issue: 4, pp: 50-54, 2011.

[29] Vidyarthi D.P, Tripathi A.K, "Maximizing reliability of distributed computing system with task allocation using simple genetic algorithm", journal of system architecture, vol: 47, issue: 6, pp. 549-554, 2001.