ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

A Skilled Algorithm for Performance Optimization of Distributed Computing System

Pankaj Saxena, Dr.Kapil Govil

Abstract*— Distributed computing is an active area of research. Distributed Computing System (DCS) is becoming bigger and more complex. DCS consists different components and objects comprising an application which can be located on different computers connected to a network. In this paper we present an, considering DCS with heterogeneous processors in order to achieve optimal results for processing time, cost and reliability by allocating the tasks to the processors, in such a way that the allocated load on each processor is balanced. The present paper shows the functionalities of algorithm in terms of performance and time-complexity. A static task allocation policy is adopted in the present paper. A best allocation of tasks leads to a best balancing of the system. As in real life situation it is assumed that the number of tasks to be allocated is more than the number of processors.

***Index Terms*— Processing Cost, Distributed Computing System (DCS), Processing Reliability, Task, Static.**

I. INTRODUCTION

In Distributed Computing System (DCS), the term Distributed means to spread or Scattered or diffuse over a network [5, 18]. A DCS is a collection of independent computers linked by a computer network that appear to the users of the system as a single coherent system. There are several advantages of DCS such as reliability, incredibility growth, sharing of resources, flexibility, and speed. Distributed computing is essential in modern computing and communications systems. Examples are large-scale networks such as the Internet. A DCS consists of multiple autonomous machines that do not share primary memory, but cooperate by sending messages over a communication network. In any DCS task allocation [1, 8, 9] is an important and interesting problem. The term "task" can be defined as the smallest, identifiable and essential piece of a job that serves as a unit of work, and as a means of differentiating between the various components of any project. It is a function to be performed. Task allocation [12, 13, 14] refers to the way that tasks are chosen, assigned, and coordinated. It is the process that results in specific processors being engaged in specific tasks, in numbers appropriate to the available processors. Task allocation [19, 20f, 21] can be static as well as dynamic. In the present paper the tasks are allocated statically because Static methods of task allocation [22, 24] have less overhead at run-time. In a static task allocation [16], the information regarding the tasks and processor attributes is assumed to be known in advance, before the execution of the tasks. The present algorithm in this paper shows the optimized results for processing time, cost [2, 7] and reliability. Processing time is the time it takes to complete a prescribed procedure. Processing cost [17] is usually a monetary valuation of resources. Processing reliability is the ability of any system to consistently perform its intended or required functions, without degradation or failure. The key factor for any successful algorithm is to check the performance by calculating the time complexity and to compare it with other existing algorithms. Time complexity expresses the relationship between the size of the input and the run time for the algorithm. This paper presents a comparison of time complexity with other existing algorithm for measuring the performance with other existing algorithms. In this paper, we investigate the problem of static task allocation of concurrent programs for DCS with processor and resource heterogeneity.

II. OBJECTIVE

The prime objective of the present paper is to give an efficient algorithm to get the optimized results in terns of Processing time, processing cost and processing reliability in such a way that after processing the several steps the algorithm should show the minimum processing time, processing cost and high processing reliability. Our objective in the present research paper is also to achieve the results of the time complexity and to compare it with other algorithms which already exist for proving our algorithm better. Objective also includes that there must be the full advantage by taking the tasks statically in completing the work and to experiment the algorithm on different inputs for giving a skilled algorithm for achieving the technical aspects.

III. NOTATIONS

n : Number of Processors
m : Number of Tasks
c : Processing Cost
PTCRM : Processing Time cost Reliability Matrix
CM : Communication Matrix
UPTCRM : Updated Processing Time Cost Reliability Matrix
MPTCRM : Merged Processing Time Cost Reliability Matrix
PTM : Processing Time Matrix
CSPTM : Column Sum Processing Time Matrix
APTM : Arranged Processing Time Matrix
PCM : Processing Cost Matrix
CSPCM : Column Sum Processing Cost Matrix
APCM : Arranged Processing Cost Matrix
PRM : Processing Reliability Matrix

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

CSPRM : Column Sum Processing Reliability Matrix
APRM : Arranged Processing Reliability Matrix
PTTAM : Processing Time Task Allocation Matrix
PCTAM : Processing Cost Task Allocation Matrix
PRTAM : Processing Reliability Task Allocation Matrix

IV. TECHNIQUE

In order to evaluate the overall optimal performance of a distributed network [15] 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 exists, where m>n. Every task contains some number of modules. For getting optimal processing time or processing cost or processing reliability for each task initially the emphasis will be on those modules of tasks which have the maximum probability of data transfer. Now, in case of time and cost the elements will be added and in case of reliability they will be multiplied. Now we have taken a matrix in which the time, cost and reliability of modules are defined and define a communication matrix by considering the communication between tasks. On the basis of highest communication we get a matrix namely MPTCRM (,,). Now from MPTCRM (,,) we can get the separate matrix for time, cost and reliability [3, 4]. Now in each matrix we will do the addition of each column and will rearrange the matrix in ascending order of their column sum. Negative and zero values will not be considered. For time and cost minimum value will be allocated and for reliability [6, 10] maximum value will be considered. Now the tasks can be allocated for getting the optimized results in terms of time, cost and reliability and Etime, Ecost and Ereliablity is calculated. The function for obtaining the overall processing time [Etime], cost [Ecost], and reliability [Ereliablity] is as follows- The function for obtaining the overall assignment time [Etime], cost [Ecost], and reliability [Ereliablity] is as follows-

$$\text{Etime} = \left[\sum_{i=1}^{n} \left\{ \sum_{j=1}^{n} ET_{ij} x_{ij} \right\} \right] \quad \text{(i)}$$

$$\text{Ecost} = \left[\sum_{i=1}^{n} \left\{ \sum_{j=1}^{n} EC_{ij} x_{ij} \right\} \right] \quad \text{(ii)}$$

$$\text{Ereliablity} = \left[\prod_{i=1}^{n} \left\{ \sum_{j=1}^{n} ER_{ij} x_{ij} \right\} \right] \quad \text{(iii)}$$

$$\text{Where, } x_{ij} = \begin{cases} \geq 1, \text{if } i^{th} \text{ task is assigned to } j^{th} \text{ processor} \\ 0, \text{Otherwise} \end{cases}$$

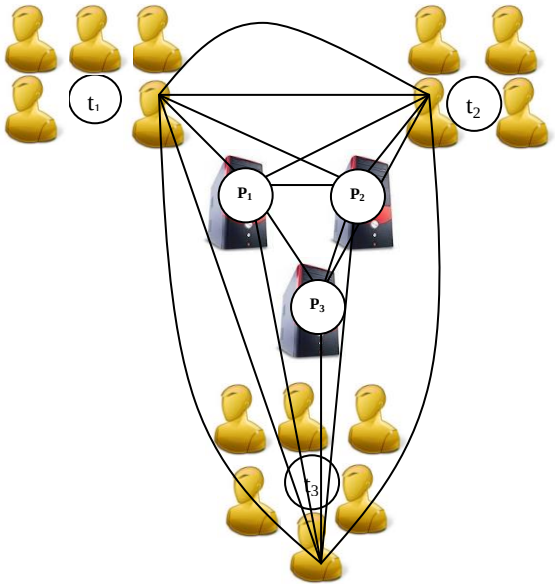
V. ALGORITHM

Step1: Start Algorithm
Step2: Read a set T= {t_1, t_2, $t_3 \ldots t_m$} of 'm' tasks.
Step3: Read a set P= {p_1, p_2, $p_3 \ldots p_n$} of 'n' processors.
Step4: Read the matrix PTCRM (,,). Select time/cost/reliability data corresponding to each task as needed of order m*n.
Step5: Input matrix CM (,,) by considering the communication time between modules of each task.
Step6: Consider each task on the basis of time or cost or reliability.
Step7: On the basis of Step 5 read the matrix UPTCRM (,,). This matrix will be derived from matrix PTCRM (,,).
Step8: Derived MPTCRM (,,) on the basis of process of merging the modules of tasks of highest communication.
Step9: From MPTCRM (,,) take the separate matrix for processing time, cost and reliability.
Step10: Compute sum of each column of PTM (,,), and store it into CSPTM (,,).
Step11: Arrange the CSPTM (,,) in ascending order of their column sum and store it into APTM (,,).
Step12: Compute addition of each column of PCM (,,) and store it into CSPCM (,,).
Step13: Arrange CSPCM (,) in ascending order of their column sum and store it into APCM (,,)
Step14: Compute sum of each column of PRM (,,) and store it into CSPRM (,,).
Step15: Arrange the CSPRM (,) in ascending order of their column sum and store it into APRM (,,).
Step16: While (all tasks! = allocated)
{
(I). Select and allocate particular tasks from APTM (,,) on processors which has the minimal Processing time.
(II). Select and allocate particular tasks from APCM (,,) on processors which has the minimal Processing cost.
(III). Select and allocate particular tasks from APRM (,,) on processors which has the maximum Processing reliability.
}
Step17: Compute the processor wise overall processing time, cost and reliability.
Step18: Display the result.
Step19: End of algorithm.

VI. IMPLEMENTATION

Let us consider a set of tasks T. This set consists three tasks t_1, t_2, and t_3. We may define it as, T= {t_1, t_2, t_3}. Now, consider the task t_1 has a set of task M_1.This set consists the five modules. We may define it as, M_1= {m_{11}, m_{12}, m_{13}, m_{14}, m_{15}}. Now, for task t_2 considers the set of task M_2.This set consists the four modules. We may define it as, M_2= {m_{21}, m_{22}, m_{23}, m_{24}}. Now, for task t_3 considers the set of task M_3.This set consists the six modules. We may define it as, M_3= {m_{31}, m_{32}, m_{33}, m_{34}, m_{35}, m_{36}}. The total number of processors are three and it can be define as, P= {p_1, p_2, p_3}. The processing time (t), cost (c) and reliability (r) of each module of every task on various processors are known and mentioned in the matrix namely, PTCRM (,,) of order 15*3 which is given below-

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013



Tasks	Modules	Processors p_1 t – c – r	p_2 t – c – r	p_3 t – c – r
	m_{11}	160 – 2600 – 0.999232	180 – 2500 – 0.999332	170 – 2800 – 0.999420
	m_{12}	140 – 2100 – 0.999764	150 – 3000 – 0.999225	120 – 2800 – 0.999385
t_1	m_{13}	130 – 2600 – 0.999123	110 – 2100 – 0.999223	130 – 2900 – 0.999784
	m_{14}	140 – 2800 – 0.999653	170 – 3000 – 0.999895	160 – 2400 – 0.999701
	m_{15}	150 – 2200 – 0.999111	110 – 2800 – 0.999327	100 – 2600 – 0.999785
	m_{21}	180 – 3000 – 0.999980	150 – 2500 – 0.999329	200 – 2500 – 0.999772
t_2	m_{22}	190 – 2400 – 0.999764	120 – 2900 – 0.999712	150 – 3000 – 0.999711
	m_{23}	160 – 2500 – 0.999555	170 – 2900 – 0.999222	155 – 2300 – 0.999806
	m_{24}	170 – 2800 – 0.999222	150 – 3000 – 0.999701	180 – 2100 – 0.999612
	m_{31}	120 – 2000 – 0.999431	100 – 2700 – 0.999944	160 – 2500 – 0.999200
	m_{32}	140 – 2600 – 0.999711	175 – 2100 – 0.999288	180 – 2700 – 0.999405
t_3	m_{33}	190 – 2700 – 0.999427	180 – 2500 – 0.999276	300 – 2900 – 0.999407
	m_{34}	160 – 2800 – 0.999908	200 – 2600 – 0.999275	210 – 2900 – 0.999500
	m_{35}	150 – 2900 – 0.999511	140 – 2900 – 0.999593	200 – 2700 – 0.999210
	m_{36}	200 – 2900 – 0.999450	110 – 2400 – 0.999282	130 – 2600 – 0.999228

The Considered Communication Time [23] Among The Modules Of Each Task Is Mentioned In The Matrices, Namely CM (,,).

For task t_1, the matrix CM (1,) of order 5*5 is as:

	m_{11}	m_{12}	m_{13}	m_{14}	m_{15}
m_{11}	0	3	9	7	1
m_{12}		0	8	5	6
m_{13}			0	3	5
m_{14}				0	4
m_{15}					0

For task t_2, the matrix CM (2,) of order 4*4 is as:

	m_{21}	m_{22}	m_{23}	m_{24}
m_{21}	0	5	8	6
m_{22}		0	8	8
m_{23}			0	7
m_{24}				0

For task t_3, the matrix CM (3,) of order 6*6 is as:

	m_{31}	m_{32}	m_{33}	m_{34}	m_{35}	m_{36}
m_{31}	0	4	7	2	1	5
m_{32}		0	8	9	8	3
m_{33}			0	6	7	7
m_{34}				0	8	4
m_{35}					0	9
m_{36}						0

Here it is considered that task t_1 is based on the constraint of execution time (one may choose the either cost or reliability constraint), task t_2 is based on the constraint of cost (one may choose the either time or reliability constraint), and task t_3 is based on the constraint of reliability (one may choose the either time or cost constraint). Hence, we can use the following form of data from matrix PTCRM (,,) i.e. the execution time for task t_1, execution cost for task t_2, and execution reliability for task t_3 and can get the matrix namely UPTCRM (,,) of order 15*3 in the following way-

Tasks	Modules	Processors p_1 t – c – r	p_2 t – c – r	p_3 t – c – r
	m_{11}	160 – … – …	180 – … – …	170 – … – …
	m_{12}	140 – … – …	150 – … – …	120 – … – …
t_1	m_{13}	130 – … – …	110 – … – …	130 – … – …
	m_{14}	140 – … – …	170 – … – …	160 – … – …
	m_{15}	150 – … – …	110 – … – …	100 – … – …
	m_{21}	… – 3000 – …	… – 2500 – …	… – 2500 – …
	m_{22}	… – 2400 – …	… – 2900 – …	… – 3000 – …
t_2	m_{23}	… – 2500 – …	… – 2900 – …	… – 2300 – …
	m_{24}	… – 2800 – …	… – 3000 – …	… – 2100 – …
	m_{31}	… – … – 0.999431	… – … – 0.999944	… – … – 0.999200
	m_{32}	… – … – 0.999711	… – … – 0.999288	… – … – 0.999405
	m_{33}	… – … – 0.999427	… – … – 0.999276	… – … – 0.999407
t_3	m_{34}	… – … – 0.999908	… – … – 0.999275	… – … – 0.999500
	m_{35}	… – … – 0.999511	… – … – 0.999593	… – … – 0.999210
	m_{36}	… – … – 0.999450	… – … – 0.999282	… – … – 0.999228

Now, the task t_1 have five modules so on the basis of highest communication the modules m_{11}&m_{13} and m_{12}&m_{15} will be fused. The task t_2 have four modules so on the basis of highest communication the modules m_{21}&m_{23} will be fused. The task t_3 have six modules so on the basis of highest communication the modules m_{32}&m_{34}, m_{31}&m_{33}, m_{35}&m_{36} will be fused. The

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

resulting matrix namely MPTCRM (,,) of order 9*3 will be as given below-

	Processors	p_1	p_2	p_3
Tasks	Modules	t – c – r	t – c – r	t – c – r
	$m_{11} * m_{13}$	290 – … – …	290 – … – …	300 – … – …
t_1	$m_{12} * m_{15}$	290 – … – …	260 – … – …	220 – … – …
	m_{14}	140 – … – …	170 – … – …	160 – … – …
	$m_{21} * m_{23}$	… – 5500 – …	… – 3400 – …	… – 4800 – …
t_2	m_{22}	… – 2400 – …	… – 2900 – …	… – 3000 – …
	m_{24}	… – 2800 – …	… – 3000 – …	… – 2100 – …
	$m_{32} * m_{34}$	… – … – 0.999619	… – … – 0.998563	… – … – 0.998905
t_3	$m_{31} * m_{33}$	… – … – 0.998858	… – … – 0.999220	… – … – 0.998607
	$m_{35} * m_{36}$	… – … – 0.998961	… – … – 0.998875	… – … – 0.998438

Now, from the MPCTR (,,) matrix we can get three matrix namely PTM (,,) of order 3*3, PCM (,,) of order 3*3 and PRM(,,) of order 3*3 which are given below-

PTM (,,) =

	$m_{11} * m_{13}$	$m_{12} * m_{15}$	m_{14}
p_1	290	290	140
p_2	290	260	170
p_3	300	220	160

PCM (,,) =

	$m_{21} * m_{23}$	m_{22}	m_{24}
p_1	5500	2400	2800
p_2	3400	2900	3000
p_3	4800	3000	2100

PRM (,,) =

	$m_{32} * m_{34}$	$m_{31} * m_{33}$	$m_{35} * m_{36}$
p_1	0.999619	0.998858	0.998961
p_2	0.998563	0.999220	0.998875
p_3	0.998905	0.998607	0.998438

Now for PTM (,,), which is for time factor, we will take the sum of each column which is given below in CSPTM (,,) of order 4*3.

CSPTM(,,) =

	$m_{11} * m_{13}$	$m_{12} * m_{15}$	m_{14}
p_1	290	290	140
p_2	290	290	170
p_3	300	220	160
$\sum$	880	770	470

On making CSPTM (,,), in ascending order of their column sum we get APTM (,,) of order 4*3 which is given below-

APTM(,,) =

	m_{14}	$m_{12} * m_{15}$	$m_{11} * m_{13}$
p_1	140	290	290
p_2	170	260	290
p_3	160	220	300
$\sum$	470	770	880

On selecting tasks which have the minimal processing time at any specified Processor, we find the allocations which are mentioned below in PTTAM (,,). Idle processor will be given the preference first.

PTTAM (,,) =

Processor	Allocated tasks	Processing time
p_1	m_{14}	140
p_2	$m_{11} * m_{13}$	290
p_3	$m_{12} * m_{15}$	220

The graphical representation is shown of PTTAM (,,) is shown in figure: 2 which is given below-

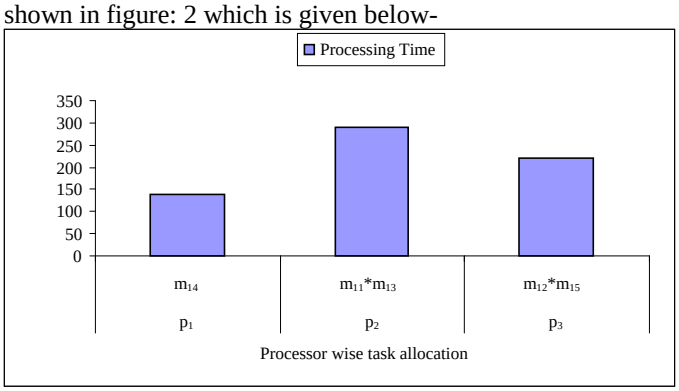


Fig 2: Processing Time task allocation

Now for PCM (,,), which is for cost factor, same as above we will take the sum of each column which is given below in CSPCM (,,) of order 4*3.

CSPCM (,,) =

	$m_{21} * m_{23}$	m_{22}	m_{24}
p_1	5500	2400	2800
p_2	3400	2900	3000
p_3	4800	3000	2100
$\sum$	13700	8300	7900

On making CSPCM (,,), in ascending order of their column sum we get APCM (,,) of order 4*3 which is given below-

APCM (,,) =

	m_{24}	m_{22}	$m_{21} * m_{23}$
p_1	2800	2400	5500
p_2	3000	2900	3400
p_3	2100	3000	4800
$\sum$	7900	8300	13700

On selecting tasks which have the minimal processing time at any specified Processor, we find the allocations which are

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

mentioned below in PCTAM (,,), idle processor will be given the preference first.

PCTAM (,,) =

Processor	Allocated tasks	Processing cost
p_1	m_{22}	2400
p_2	$m_{21} * m_{23}$	3400
p_3	m_{24}	2100

The graphical representation is shown of PCTAM (,,) is shown in figure: 3 which is given below-

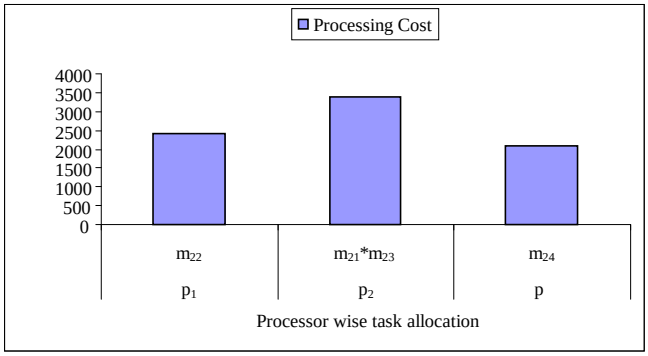


Fig 3: Processing Cost Task Allocation

Now for PRM (,,), which is for reliability factor, same as above we will take the sum of each column which is given below in CSPRM(,,) of order 4*3-

		$m_{32} * m_{34}$	$m_{31} * m_{33}$	$m_{35} * m_{36}$
CSPRM(,,) =	p_1	0.999619	0.998858	0.998961
	p_2	0.998563	0.999220	0.998875
	p_3	0.998905	0.998607	0.998438
	$\prod$	0.997089	0.996688	0.996278

On making CSPRM (,,), in ascending order of their column sum we get APRM (,,) of order 4*3 which is given below-

		$m_{32} * m_{34}$	$m_{31} * m_{33}$	$m_{35} * m_{36}$
APRM(„) =	p_1	0.999619	0.998858	0.998961
	p_2	0.998563	0.999220	0.998875
	p_3	0.998905	0.998607	0.998438
	$\prod$	0.997089	0.996688	0.996278

On selecting tasks which have the maximal processing reliability at any specified Processor, we find the allocations which are mentioned below in PRTAM (,,), idle processor will be given the preference first.

PRTAM (,,) =

Processor	Allocated tasks	Processing reliablity
p_1	$m_{32} * m_{34}$	0.999619
p_2	$m_{31} * m_{33}$	0.999220
p_3	$m_{35} * m_{36}$	0.998438

The graphical representation is shown of PRTAM (,,) is shown in figure: 4 which is given below-

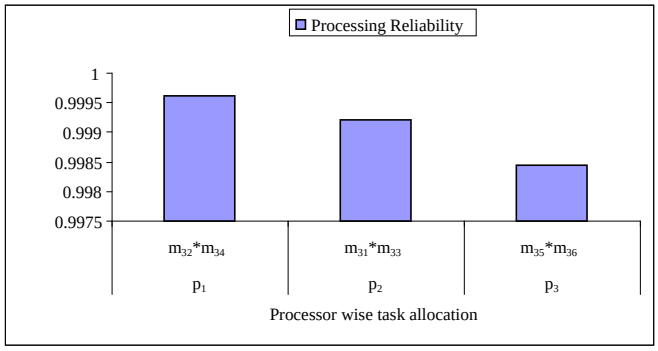


Fig 4: Processing Reliability Task Allocation

Now, From PTTAM (,) the overall processing time can be given by the following way-

Processors	Task allocation	Processing time
p_1	m_{14}	140
p_2	$m_{11} * m_{13}$	290
p_3	$m_{12} * m_{15}$	220
Overall Processing Time	:	650

From PCTAM (,,) the overall processing cost can be given by the following way-

Processors	Task allocation	Processing Cost
p_1	m_{22}	2400
p_2	$m_{21} * m_{23}$	3400
p_3	m_{24}	2100
Overall Processing Cost	:	7900

From PRTAM (,,) the overall processing reliability can be given by the following way-

Processor	Task allocation	Processing reliablity
p_1	$m_{32} * m_{34}$	0.999619
p_2	$m_{31} * m_{33}$	0.999220
p_3	$m_{35} * m_{36}$	0.998438
Overall Processing Reliablity	:	0.997279

V. CONCLUSION

From the above results for overall processing time, processing cost and processing reliability the conclusion can be present in a tabular form for optimal Etime, Ecost and Ereliablity which is given below-

	Processors			Optimal	Optimal	Optimal
Tasks	p_1	p_2	p_3	Etime	Ecost	Ereliablity
t_1	m_{14}	$m_{11} * m_{13}$	$m_{12} * m_{15}$	650	…	…
t_2	m_{22}	$m_{21} * m_{23}$	m_{24}	…	7900	…
t_3	$m_{32} * m_{34}$	$m_{31} * m_{33}$	$m_{35} * m_{36}$	…	…	0.997279

The analysis of any algorithm [11] specifically emphasis on

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

time complexity. The time complexity is a measure of the amount of time required to execute an algorithm. Time complexity expresses the relationship between the size of the input and the run time for the algorithm. It is a function of input size 'n'.The time complexity of the above mentioned algorithm is O (mn). By taking several input examples, the above algorithm gives the results mentioned below-

No.of Processors(n)	No.of Tasks(m)	Optimal Results
3	4	12
3	5	15
3	6	18
3	7	21
4	5	20
4	6	24
4	7	28
4	8	32
5	6	30
5	7	35
5	8	40
5	9	45

The graphical representation of the above results are shown by figure 5, 6, and 7as given below-

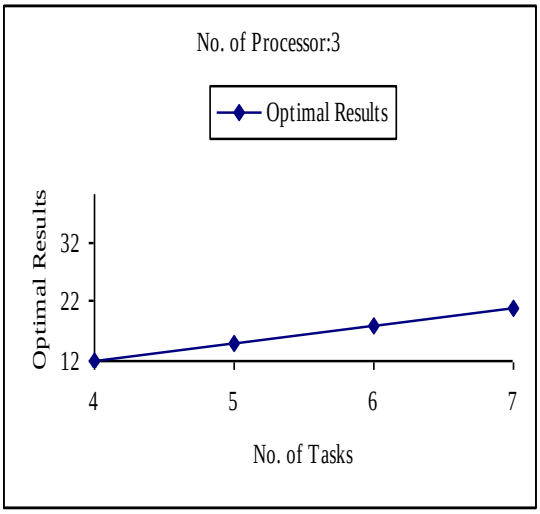


FIG 5: Processor Wise Complexity

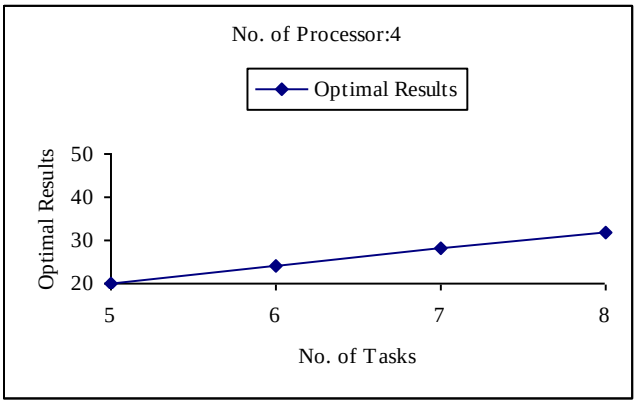


FIG 6: Processor wise Complexity

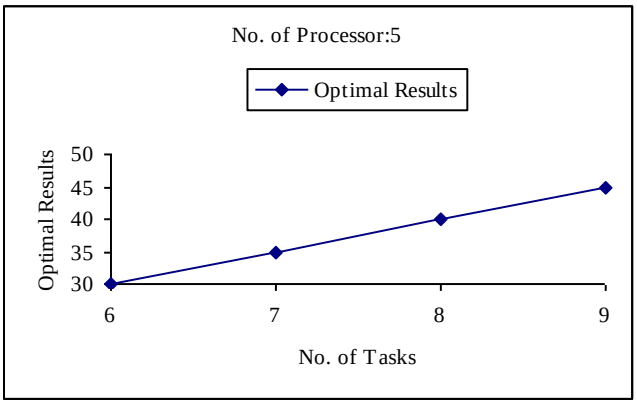


FIG 7: Processor Wise Complexity

The performance of the suggested algorithm is compared with the algorithm suggested by algorithm [23] as the given table shows the comparison of time complexity between algorithm [23] and presented algorithm-

Processor(n)	Tasks(m)	Time Complexity of algorithm[12] $O(mn^3 \log n)$	time complexity of present algorithm O(mn)
3	4	51	12
3	5	64	15
3	6	77	18
3	7	90	21
4	5	192	20
4	6	231	24
4	7	269	28
4	8	308	32
5	6	524	30
5	7	611	35
5	8	699	40
5	9	786	45

From the above table it is clear that algorithm proposed by us is much better for the optimal allocation of tasks for enhancing the performance of distributed computing system. The graphical representation is given below between algorithm [23] and present algorithm by figure 8, 9, and 10-

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

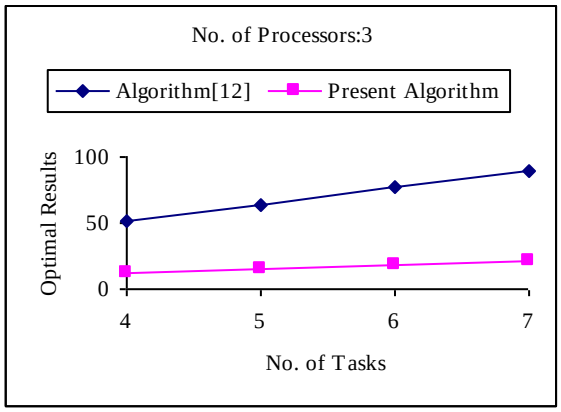


Fig 8: Comparison Graph

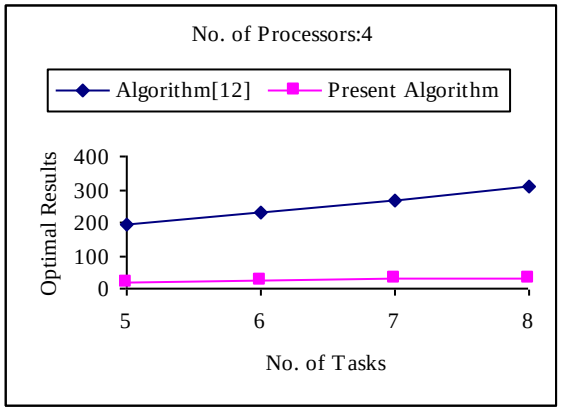


Fig 9: Comparison Graph

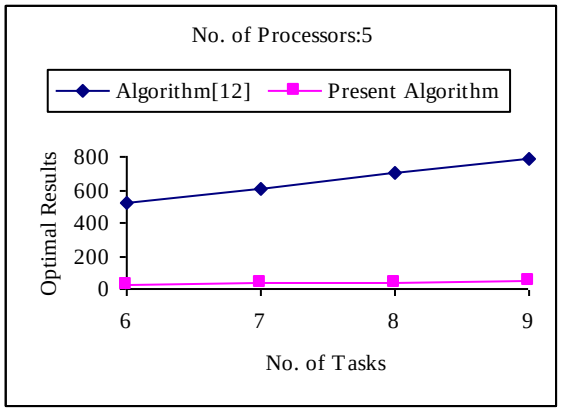


Fig 10: Comparison Graph

REFERENCES

[1] Ahmed Younes. Hamed, "Task Allocation for Minimizing Cost of Distributed Computing Systems Using Genetic Algorithms," International Journal of Advanced Research in Computer Science and Software Engineering vol.2, pp. 202-209, September 2012.

[2] Anju Khandelwal, "Optimal Execution Cost of Distributed System: Through Clustering," International Journal of Engineering Science and Technology (IJEST), vol. 3, pp.2320-2328, March 2011.

[3] A. Y. Hamed, "Task Allocation for Maximizing Reliability of Distributed Computing Systems Using Genetic Algorithms," International Journal of Computer Networks and Wireless Communications, vol.2, pp.560-569, October 2012.

[4] Ajit Kimar Verma, and Mangesh Trimbak Tamhankar, "Reliability Based optimal Task Allocation in Distributed Database Management Systems," IEEE Transactions on reliability, vol. 46, pp. 453-459, 1997.

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

[6] Anurag Raii, and Vikram Kapoor, "Reliable Clustering Model for Enhancing Processors Throughput in Distributed Computing System," International Journal of Computer Applications, vol: 38, pp: 47-50, 2012.

[7] Bo Yang, Huajun Hu, and Suchang Guo, "Cost-oriented task allocation and hardware redundancy policies in heterogeneous distributed computing systems considering software reliability," Journal of Computers & Industrial Engineering, vol. 56, pp. 1687-1696, 2009.

[8] Braun Tracy D, Siegel Howard jay, maciejewski Anthony A, and Hong ye, "Static resource allocation for heterogeneous computing environments with tasks having dependencies, priorities, deadlines, and multiple versions," Journal of parallel and distributed computing, vol. 68, pp. 1504-1516, 2008.

[9] 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.

[10] 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.

[11] D. Coit, and A. Smith, "Reliability optimization of series-parallel systems using genetic algorithm," IEEE Tran on. Reliability, vol.45, pp. 254-266, 1996.

[12] Gamal Attiya, and Yskandar Hama, "Task allocation for maximizing reliability of distributed systems: A simulated annealing approach," Journal of Parallel and Distributed Computing, vol. 66, pp. 1259-1266, 2006.

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

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

[15] Jorge E. Pezoa, Sagar Dhakal and Majeed M. Hayat, "Maximizing Service Reliability in Distributed Computing Systems with Random Node Failures: Theory and Implementation," IEEE transactions on parallel and distributed systems, vol. 21, pp.1531-1544, October 2010.

[16] Jia-Ping Wang and M. Shatz, "Task Allocation for Maximizing Reliability of Distributed Computer Systems Sol," IEEE Transactions on computers, vol. 41, pp.1156-1168, September 1992.

[17] Keren A, and Barak A, "opportunity cost algorithms for reduction of I/O and inter process communication overhead in a computing cluster," IEEE transaction on parallel and distributed systems, vol. 14, pp.39-50, 2003.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 2, Issue 9, March 2013

[18] Marwa Shouman, Gamal Attiya, and 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, pp.18-24, 2011.

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

[20] Manisha Sharma, Harendra Kumar, and 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, pp.50-55, 2012.

[21] P. K. Yadav, M. P. Singh, and Kuldeep Sharma, "An Optimal Task Allocation Model for System Cost Analysis in Heterogeneous Distributed Computing Systems: A Heuristic Approach," International Journal of Computer Applications, vol.28 pp.0975-8887, August 2011.

[22] Pankaj Saxena, Kapil Govil, Gaurav Saxena, Saurabh Kumar and Neha Agrawal, "an algorithmic approach and comparative analysis of task assignment to processor for achieving time efficiency in process completion," International Journal of Applied Engineering and Technology, vol. 2, pp.114-119, 2012.

[23] Xibo Jin, Fa Zhang, Ying Song, Liya Fan and Zhiyong Liu, "Energy efficient scheduling with time and processor eligibility restrictions," arXiv: 1301.4131v1 [cs.DS] 17 Jan 2013.

[24] Zubair khan, ravinder singh, and Jahangir alam, "task allocation using fuzzy inference in parallel and distributed system," Journal of Information and Operations Management, vol. 3, pp.322-326, 2012.