To Propose a Weight based Scheme to Reduce Faults in DCS

Harjot Kaur
Student, Department of ECE
BBSBEC, Fatehgarh Sahib, Punjab

Raju Sharma
Assistant Professor, Department of ECE
BBSBEC, Fatehgarh Sahib, Punjab

Abstract: Distributed systems play an important role on achieving good performance and high system utilization. The goal of a job scheduling system is to efficiently manage the distributed computing power of workstations, servers, and supercomputers in order to maximize job throughput and system utilization. There are many issues of distributed computing system which are discussed in this paper in brief. The main aim of the paper is to focus on fault tolerance and recover fault with less processing time. The proposed algorithm is assign tasks to other nodes only when master node moves from its original position. The major problem in this architecture is task scheduling, if one slave node get failed the task allocated by master node will not get completed and fault occurred. In this work, we have worked on technique which helps to reduce fault tolerance of the system and increase performance of the system.

Keywords: Distributed systems. Task allocation, job scheduling, and scalability

I. INTRODUCTION

The distributed system is a collection of independent computers that appears to the users as a single coherent system. It can be of two types either homogenous or heterogeneous. A heterogeneous distributed computing system is that where random node can fail permanently. A homogenous distributing computing system is that which shares local memory. Because the DCS is heterogeneous, so its various nodes have different hardware and software characteristics. The different components of the application also have various hardware and
software requirements. A distributed system connected by local networks and physically connected with each others. Distributed computing utilizes a network of many computers, each accomplishing a portion of an overall task, to achieve a computational result much more quickly than with a single computer. A computer program that runs on distributed system is known distributed program. The
process of writing such types of languages is called distributed programming [4]. First of all each computational entity has local memory. The entities communicate with each others with the help of message passing. Second the system has to tolerate failures in individual computers.The system structure and links may changes during the execution of the distributed programs. Each system is only aware about the input of the system. Resource sharing is the capability to use any hardware, software or data anywhere in the system. Resources in a distributed system, dissimilar the centralized one, are physically encapsulated within one of the computers and can only be accessed from others by communication. Openness is apprehensive with extensions and improvements of distributed systems. New components have to be included with presented components so that the added functionality becomes accessible from the distributed system as a whole [6].

1.1 Design Issues of Distributed System: A Distributed operating system must be designed to provide all the advantages of a distributed system to its users. There are some design issues which are as follows:

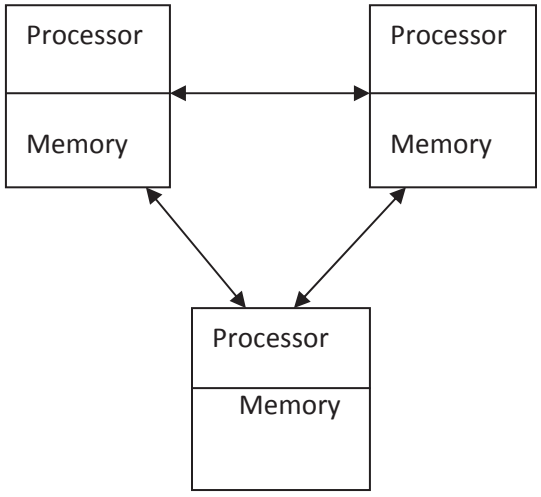

Fig.1 Distributed Computing Systems

1. Flexibility: The distributed system should be flexible so that modifications and enhancement can be done easily by the users.

*2. Scalability:*System should be designed in this manner that it is easily coping up with it increase growth of the system. It should avoid central algorithms and central entities. It should be perform most of the operation at the client work station.

3. Security: In order that the users can trust the system and rely on it, the various resources of a computer system must be protected against destruction and unauthorized access. Enforcing security in a distributed system is more difficult than in a centralized system because of the lack of a single point of control and the use of insecure networks for data communication.

*4. Fault Tolerance:*The system must be resistance to faults. In future if any fault may occur it doesn't degrade its performance.

II. REVIEW OF LITERATURE

In this paper [1] they addressed the distributed tracking control problem for multi-agent systems with heterogeneous uncertainties and a leader whose control input might be nonzero and not available to any follower. Based on the relative states of neighboring agents, both distributed continuous static and adaptive controllers have been designed to guarantee the uniform ultimate boundedness of the tracking error for each follower. A sufficient condition for the existence of these distributed controllers is that each agent is stabilizable. In this paper [2] they represented Mobile Agent technology promises to be a powerful of the agent, to know where the agent is and what is it mechanism to improve the flexibility and doing. Mobile agent systems must also provide an customizability of applications with its ability to additional feature for the security for the agent from dynamically deploy application components across the malicious host and the security of the host from a network. But none of the present mobile agent is malicious agents. The architecture proposed in this paper prototype systems satisfy all the requirements to address the above issues can be used to extend to provide a secure and reliable architecture, suitable for features of the existing systems. In this paper [3] they presented in this paper a service for faults tolerance which includes an algorithm of modeling in group in the Ad hoc networks for then applying the tolerance to the faults by replication which is based primarily on the prediction. The definite algorithm forms the groups on the basis of number of neighbors and energy level of nodes. The network after clustering has a hierarchical structure of two levels with a leader for each group and a super leader (generic) for all networks. The formed groups are open, dynamic, a disjoined, explicit, and

allowing point-to-point communication of group. The service of faults tolerance applied is composed of four sub services to knowing (clustering, decision, replication by prediction and consistency making it possible to better manage the network and which integrates the functionalities necessary for a better availability of the data. Our contributions takes account of the characteristics of terminal mobile with an aim of reducing the maximum loss of information and reduce the consumption of their critical resources which is energy. For the futures works: we wish to implement our service of faults tolerance in a simulator of network Ad hoc such as NS2 or GloMo Sim and improve our algorithm used in sub service of clustering by taking into account of intention of nodes and to integrate into our simulator various protocols of routings for the Ad hoc networks. In this paper [4] they defined problem of the task allocation in distributed computing system is to need to allocate a number of tasks to different processors for execution. The paper deals with the problem of task allocation in heterogeneous distributed computing systems with the goal of maximizing the system reliability. They present a genetic algorithm to obtain the optimal solution for this problem. In the performance of the algorithm they consider more than one parameter such as the number of tasks, the number of processors, and task interaction density of applications. The experimental results illustrate the effectiveness of this algorithm over conventional algorithms.

III. FAULT TOLERANCE IN DISTRIBUTED COMPUTING SYSTEM

Distributing computing is a computational system in which software and hardware infrastructure provides consistence, dependable and inexpensive to accesses high end computations. An imperfect system due to some reasons can cause some damages. A task which is working on real time distributed system should be achievable, dependable and scalable [6]. The real time distributed systems like grid, robotics, nuclear air traffic control systems etc. are highly responsible on deadline. Any mistake in real time distributed system can cause a system into collapse if not properly detected and recovered at time. Fault-tolerance is the important method which is often used to continue reliability in these systems. By applying extra hardware like processors, resource, communication links hardware fault tolerance can be achieved. In software fault tolerance tasks, to deal with faults messages are added into the system. Distributed computing is different from traditionally distributed system [7]. Fault Tolerance is important method in distributed computing because nodes are distributed geographically in this system under different geographically domains throughout the web wide. The most difficult task in distributed computing is design of fault tolerant is to verify that all its reliability requirements are meet [7]. Fault Tolerance can be achieved with the help of two ways. These ways are as follow:

1. Recovery
2. Redundancy

A good fault- tolerant system design requires a careful study of failures causes of failures and system responses to failures. Such learning should be approved out in aspect before the design start and have to remain part of the design process [9].Planning to keep away from failures is most important. A designer must examine the situation and decide the failures that must be tolerated to achieve the preferred level of dependability. To optimize fault tolerance, it is important to calculate approximately actual failure rate for each possible failure.

IV. PROPOSED METHODOLOGY

With the dramatic increase of the scales of today's distributed systems, it is urgent to develop efficient job schedulers. The number of users of distributed systems and networks considerably increases with the increasing complexity of their services and policies, system administrators attempt to ensure high quality of services each user requires by maximizing utilization of system resources. To achieve this goal, correct, real-time and efficient management and monitoring mechanisms are essential for the systems. But, as the infrastructures of the systems rapidly scale up, a huge amount of monitoring information is produced by a larger number of managed nodes and resources and so the complexity of network monitoring function becomes extremely high. Thus, mobile agent-based monitoring mechanisms have actively been developed to monitor these large scale and dynamic distributed

networked systems adaptively and efficiently. The proposed algorithm is assign tasks to other nodes only when master node moves from its original position. The major problem in this architecture is task scheduling, if one slave node get failed the task allocated by master node will not get completed and fault occurred. In this work, we will work on technique based upon weight which helps to reduce fault tolerance of the system and increase performance of the system.

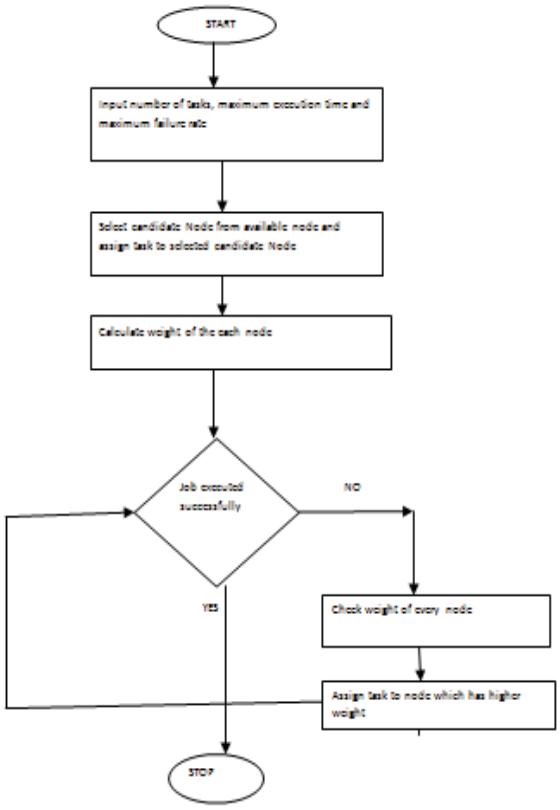

Fig. 2: Flowchart of methodology

V. EXPERIMENTAL RESULTS

The entire proposed work has implemented on NS2 platform.

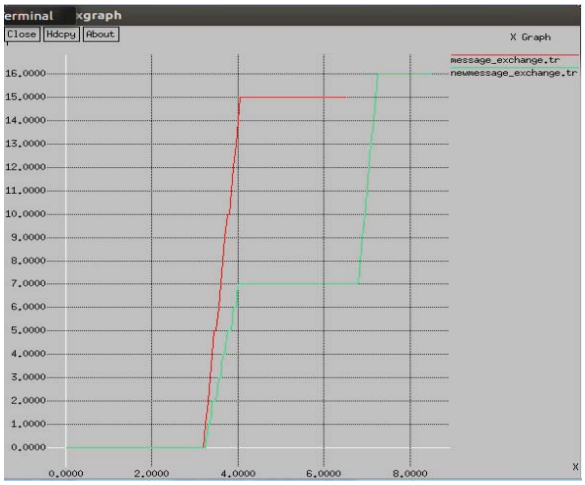

Fig.3: Message Exchange Graph

In this fig. message exchange parameter is considered. X axis represent time takes to exchange messages and Y-axis represents no. of packets. As illustrate in figure red line displays existing technique's performance. In this technique message are exchanged less due to mobility of the node. Due to this performance of the network degrade. Similarly, Green line represents proposed technique which exchange more messages in the network as compared to the existing technique. In this way proposed technique is better than the existing technique.

Fig.4: Packet Overhead

In this fig. packet overhead parameter is considered. X axis represent nodes takes to exchange messages and Y-axis represents no. of packets. As illustrate in figure red line displays existing technique's performance. In this technique packet overhead is more due to mobility of the node. Due to this performance of the network degrade. Similarly, Green line represents proposed technique which has fewer packets overhead in the network as compared to the existing technique. In this way proposed technique is better than the existing technique.

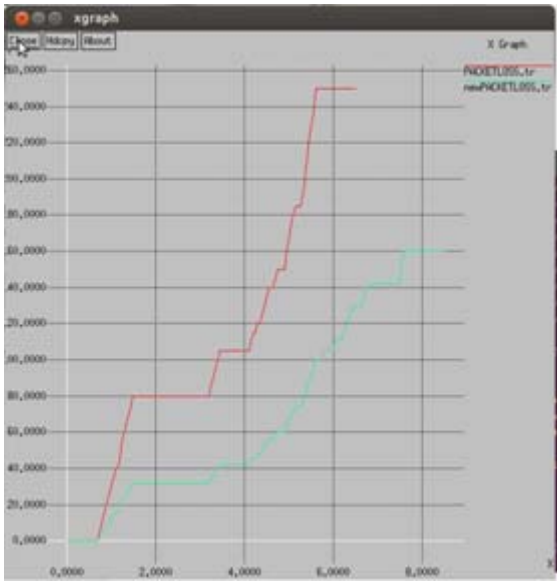

Fig. 5: Packet loss

In this fig. packet packet loss parameter is considered. X axis represent time takes to exchange number of packets and Y-axis represents no. of packets. As illustrate in figure red line displays existing technique's performance. In this technique packet loss is more due to mobility of the node. Due to this performance of the network degrade. Similarly, Green line represents proposed technique which has fewer packet loss in the network as compared to the existing technique. In this way proposed technique is better than the existing technique.

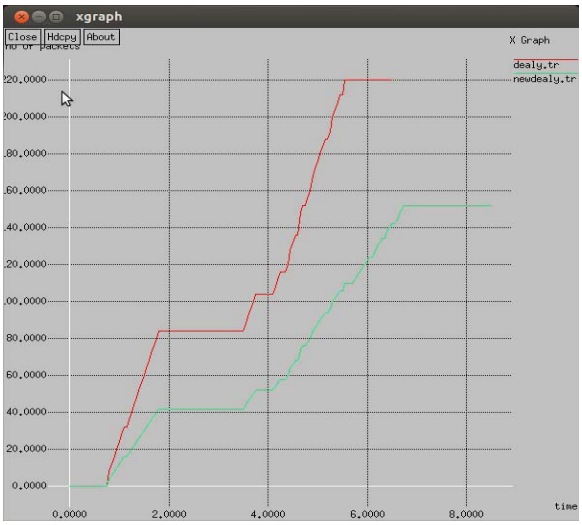

Fig. 6: Delay

In this fig. delay is considered. X axis represent time and Y-axis represents no. of packets. As illustrate in figure red line displays existing technique's performance. In this technique delay is more due to mobility of the node. Due to this performance of the network degrade. Similarly, Green line represents proposed technique which has less delay in the network as compared to the existing technique due to removal of fault in the network. In this way proposed technique is better than the existing technique.

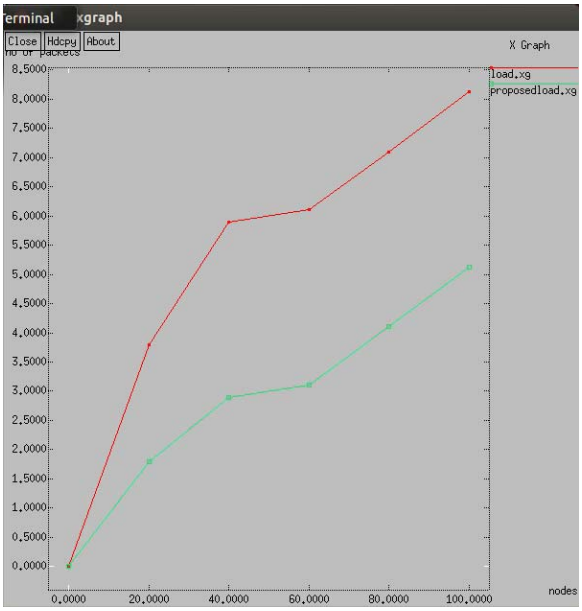

Fig.7: Network load

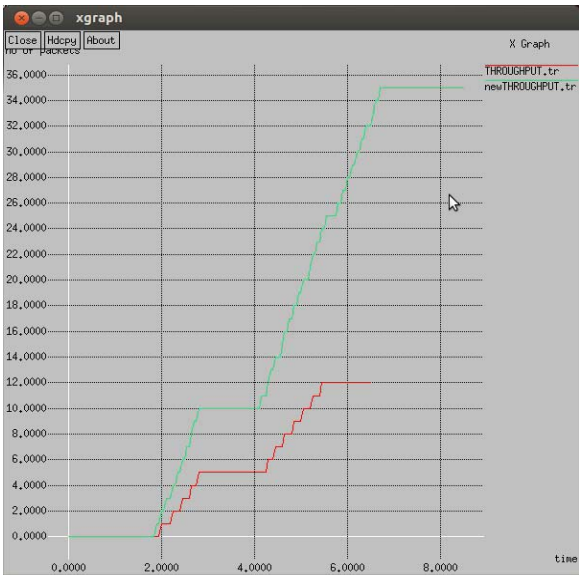

Fig. 8: Throughput

VI. CONCLUSION

As the distributed network is defined as a relatively dense of collection of mobile entities connected by a wireless link, without any administration or fixed support. In this network no central authority is present due to which the network disconnection is very frequent between the mobile nodes. Due to above reasons chances of errors in the mobile distributed network is very high. The load is equally divided among the mobile node to enhance the network efficiency and to reduce the task execution time. When the load is not equally divided among the mobile nodes, chance of error occurrences will be increased. The approach of fault tolerance is required to reduce the number of error rates in mobile distributed network. The task allocation among the mobile nodes is done with the use of task allocation modal. In this thesis, novel technique has been proposed which reduces the fault detection time in the network and reduces the resource consumption to execute the allocated tasks using weight based technique. The

proposed algorithm is based on the failure rate, minimum execution and time taken by the master node scheme for fault recovery and concurrent execution of processes for the process execution. This technique leads to reduce in processing time and reduce in energy consumption.

REFERENCES

[1] Zhongkui Li and Zhisheng Duan, "Distributed Tracking Control of Multi-Agent Systems with Heterogeneous Uncertainties" , 10th IEEE International Conference on Control and Automation (ICCA) Hangzhou, China, June 12-14, 2013

[2] Sreedevi R.N, Geeta U.N, U.P.Kulkarni , A.R.Yardi, "Enhancing Mobile Agent Applications with Security and Fault Tolerant Capabilities, 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009

[3] Asma Insaf Djebbar, Ghalem Belalem , "Modeling by groups for faults tolerance based on multi agent systems", IEEE,2010

[4] A. Y. Hamed, "Task Allocation for Maximizing Reliability of Distributed Computing Systems Using Genetic Algorithms", International Journal of Computer Networks and Wireless Communications (IJCNWC), ISSN: 2250-3501 Vol.2, No5, October 2012

[5] P K Yadav, M P Singh and Kuldeep Sharma. Article, "An Optimal Task Allocation Model for System Cost Analysis in Heterogeneous Distributed Computing Systems", A Heuristic Approach. *Interntional Journal of Computer Applications* 28(4):30-37, August 2011

[6] Shilpa Gambhir ,Er. Sonia Goyal, "Reliable Task Allocation in Distributed Mobile Computing System with random node movement: Replication and Load Sharing Approach", International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE) Volume 3 , Issue 6, June 2014

[7] Qin&Ma Kng, Hong He, Hui& Min Song, Rong Deng, "Task allocation for maximizing Reliability of distributed Computing System using Honeybee mating Optimization" ,The Journal of Systems and software, Vol.83,issue 2, 2010

[8] A.A. Elsadek & B. E. Wells, "A heuristic model for task allocation in heterogeneous distributed computing systems," International journal of computers and there applications, Vol.6, No.1, March 1999. Pp. 0&35.

[9] V.M. Lo, "Heuristic Algorithms for Task assignment in distributed systems," IEEE Transactions on computers, Vol.37. No. 11, pp. 1384& 1397, November , 2010.

[10] Vinod Kumar Yadav, Mahendra Pratap Yadav and Dharmendra Kumar Yadav , "Reliable Task Allocation in Heterogeneous Distributed System with Random Node Failure: Load Sharing Approach, International Conference of Computing Science, 2012

[11] S. Ghosh, R. Melhem, D. Mosse, "Fault-tolerance through scheduling of a periodic tasks in hard real-time multiprocessor systems", IEEE Transactions on Parallel and Distributed Systems, Vol.8 (3), 1997, pp.272-284.

[12] G. Rodrıguez, M. J. Martın, P. Gonzalez, and J. Tourino. "Controller/Precompiler for Portable Checkpointing", IEICE Transactions on Information and Systems, Feb. 2006, pp.408–417.