Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 1, Number 6; October, 2014 pp. 10-12
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/jbaer.html

Fault Tolerance in Distributed Mobile Computing-A Review

Anupama Padha[1], Meenakshi Sharma[2]

[1,2]Punjab Technical University, Sri Sai college of Engg and Technology, Badhani

Abstract: **Distributed computing is an independent and heterogeneous system in nature. 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. So there is a problem of fault tolerance in distributed system when one of the working node moves. In this paper we studied about fault tolerance in distributed system. A novel technique will be proposed to overcome this problem.**

Keywords: **Computing, Fault Tolerance, openness, reliability**

1. INTRODUCTION

A distributed system is a collection of independent computers that appears users as a single coherent system. A heterogeneous distributed computing system is that where random node can fail permanently. 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 la nguages is called distributed programming [14]. Grid computing and Cluster computing are types of distributed computing systems. A Distributed system consists of a group of independent computers associated through a network and sharing middleware which enables computers to organize their behavior and to share the property of the system so that users identify the system as a single, in corporated computing facility [15]. There are many properties of distributed computing system. 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 [16]. In this section we described about distributed computing system. In section 2nd we will do literature survey. In section 3rd we will discussed about it.

2. REVIEW OF LITERATURE

In paper [1] Vinod Kumar Yadav had tried to solve the problem of maximizing reliability of heterogeneous distributed computing system where random node can fail permanently. 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. Firstly, they determine the candidate nodes for tasks that can satisfy to its requirements. Then they utilize the load sharing policies for handling the nodes failure as well as maximizing the service reliability of DCS. They have undertaken a two-phase hybrid approach to analyze the service reliability of heterogeneous DCS in the presence of communication and node uncertainty. On the basis of requirements of tasks, capabilities of processors and communication links, we determined the candidate processors. In second-phase we used the concepts of load sharing to handle the execution of tasks. They assigned each tasks to those processor that is most reliable and cost effective for this tasks. Instead of that, if any processor failed before executing the tasks assigned on it, it will transfer reaming tasks to next more reliable and cost effective processor. Repeat this phase until all the tasks got executed and compared with those derived by load sharing approach. The simulation result shows that, in most of the cases this hybrid solution gives the more cost effective results than load sharing approach. For a small test case of eight tasks, it improved the performance up to 20% from load sharing solutions. Performance graph shows as the number of tasks increases, the performance of hybrid approach will also increases.

In this paper [2], **Tome Dimovski, Pece Mitrevski,** they proposed a Connection Fault- Tolerant Model for mobile environment which considers two communication scenarios first is when MHs can connect to the fixed network through MSS, and the second when MHs cannot connect to the fixed network. We presented a Decision Algorithm which is responsible for making a decision for a MH when corresponding MH-Ag cannot communicate with its MH for a defined period of time. The CFT model reduces the blocking time of resources at the fixed devices, provides fast recovery from connection failures owing to mobility of mobile devices and increases the number of committed mobile transactions.

In this paper Rajwinder Singh et.al [3] they present mobile agent based fault prevention and detection technique where the team of mobile agents monitor each host in mobile agent based system. This research focuses on building an automatic, adaptive and predictive determining policy where critical host agents are identified in advance by monitoring agents, to avoid their failures. This paper presented a new approach to make mobile agent systems reliable. They proposed an approach to introduce fault tolerance in multi agent system through check pointing based on updating of weights from time to time while calculating the dependence of hosts. From experimental results it can be safely inferred that the proposed monitoring technique for multi agent distributed application may effectively increase system's fault tolerance beside effective recognition of vulnerabilities in system. In the future, they intend to work out a more formal model of the quantity of dependence and incorporate other parameters to gauge the efficiency of the model in accurately measuring host vulnerability.

In this paper **Asma Insaf Djebbar [4]** they presented that the mobile Ad hoc networks are distributed environments characterized by a high mobility and limited battery resources. In these networks, mobiles nodes are subject to many errors. In this paper, they present their approach of modeling by groups for faults tolerance based in MAS, which predicts a problem and provide decisions in relation to critical nodes. Their work contributes to the resolution of two points. First, they propose an algorithm for modeling by groups in wireless network Ad hoc. Secondly, they study the fault tolerance by prediction of disconnection and partition in network; therefore we provide an approach which distributes efficiently the information in the network by selecting some objects of the network to be duplicates of information.

In this paper [5] , Rajwinder Singh et.al they proposes a novel parallel check pointing algorithm antecedence graph approach for achieving fault tolerance in mobile agent systems. By recording the dependency relation among mobile agents in antecedence graphs and check pointing them to stable storage during the normal computation message transmission, the proposed algorithm can reduce the time latency for a global check pointing procedure significantly. Furthermore, it only forces the minimum number of MAs to take their checkpoints and minimizes the number of blocked mobile agents during identifying, which improves the system performance compared with previous graph based approaches. The quantitative analysis and numerical results reveal that the proposed algorithm has better performance than existing ones and the overheads for proposed scheme are significantly low. These advantages enhance the applicability of fault tolerance approach using antecedence graphs for mobile agent systems. The future scope of the work includes comparison of the proposed schemes with other existing fault tolerant schemes. Implementation by use of other check pointing schemes can also be done to improve the proposed scheme in order to further enhance the execution time and recovery time. In future, work could be done for integrating graph based and non-graph-based schemes to achieve high level of fault tolerance for making real life, mobile agent-based applications more reliable and fault tolerant.

3. FAULT TOLERANCE IN DISTRIBUTED SYSTEMS

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. 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. 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.

4. PROBLEM FORMULATION

Mobile agent technology has become a new paradigm for distributed real-time systems because of their inherent advantages. The Distributed systems can reduce the load on the central authority. The central authority can distribute the task to various other mobile systems. This approach will enhance the network throughput, reduce execution time and reduce battery consumption. The network is the mobile network and network's topology will change suddenly. As the mobile network is defined as a relatively dense of collection of mobile entities connected by a wireless link, without any administration or fixed support. In the mobile 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 task allocation modal On the basis of capacities of processors and communication links, we allocate the tasks among processors. Failure problem can be solved by task redundancy. Task redundancy is provided by backup system that is attached with each node of the DCS. Here, it is noted that backup system does not provide service to any tasks. In case of node failure backup system will perform the following operations: 1) multicast a failure notice (FN) to alert the candidate nodes about the change in the number of functioning nodes; 2) reallocate all the unfinished tasks among those candidate nodes perceived to be functioning; When any node fails or when load on any node will increase, back up node will come into existence. The backup node will execute the task allocation algorithms to balance load between the available mobile nodes. In the existing modal, we need efficient task allocation algorithms and we need to define the certain parameters on the basis of which backup node will identify that on which node load is increased. In case any node which has task moves from its position then its task will assign to the previous node. So in this case time process increase and fault tolerance is maximum. To overcome this, novel technique will be proposed so that fault tolerance improves than existing technique.

5. CONCLUSION

Distributed Computing system is heterogeneous in nature. Distributed system consists of a group of independent computers associated through a network and sharing middleware which enables computers to organize their behavior. In this paper we focused on the fault tolerance in distributed computing which is one of the major issues in distributed computing. When one node moves from the system its performance degrades. So to overcome this problem a novel technique will be proposed which will increase performance and throughput of the system.

REFERENCES

[1] 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

[2] Tome Dimovski, Pece Mitrevski, "Connection Fault-Tolerant Model for Distributed Transaction Processing in Mobile Computing Environment" ITI 2011 33rd Int. Conf. on Information Technology Interfaces, June 27-30, 2011, Cavtat, Croatia

[3] Rajwinder Singh, Mayank Dave, "Using Host Criticalities for Fault Tolerance in Mobile Agent Systems, 2nd IEEE International Conference on Parallel, Distributed and Grid Computing, 2012

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

[5] Rajwinder Singh and Mayank Dave, Senior Member, "Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems", IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 2, FEBRUARY 2013

[6] Yang, 1. Cao and W. Wu, "CIC: An integrated approach to checkpointing in mobile agent systems", Proceedings of the Second IEEE International Conference on Semantics, Knowledge and Grid,pp. 1-6, 2006.

[7] W. Qu and H. Shen, "Analysis of mobile agents fault-tolerant behavior", Proceedings of IEEEIWIC/ACM international conference on intelligent agent technology, pp. 377 - 380, 2004.

[8] Bahi, Jacques, Couturier, Raphael and Vernier, Flavien. Synchronous distributed load balancing on dynamic networks, Journal of Parallel and Distributed Computing, Elsevier Inc., Vol. 65, Issue 11, 1397 – 1405, 2005.

[9] A. Liotta , G. Pavlou and G. Knight, Exploiting Agent Mobility for Largescale Network Monitoring, IEEE Network, 2002, 7-15.

[10] S. Kwon and J. Choi, An Agent-based Adaptive Monitoring System, Lecture Notes In Artificial Intelligence, 4088, 2006, 672-677.

[11] J. Philippe, M. Flatin and S. Znaty, Two Taxonomies of Distributed Network and System Management Paradigms, Emerging Trends and Challenges in Network Management, 2000.

[12] G. Susilo, A. Bieszczad and B. Pagurek, Infrastructure for Advanced Network Management based on Mobile Code, In Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'98), 1998, 322-333.

[13] K. Park, "A fault-tolerant mobile agent model in replicated secure services", Springer, Proceedings of International Conference Computational Science and Its Applications, Vol. 3043, pp. 500-509,2004

[14] S.M. Shatz, J. P. Wang, and M.Goto, "Task allocation for maximizing reliability of distributed computing system", IEEE trans. computers, vol, 41, no. 9, pp. 1156 - 1168,1992.

[15] A.K. Verma, M.T. Tamhankar, Reliability-based optimal task-allocation in distributed-database management systems, IEEE Trans. Reliab. 46 (1997) 452-459.

[16] S.M. Shatz, J. P. Wang, and M. Goto, "Task allocation for maximizing reliability of distributed computing system", IEEE trans. computers, vol, 41, no. 9, pp. 1156 1168,1992.