International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 3, Issue 6, June 2014

Reliable Task Allocation inDistributed Mobile Computing System with random node movement: Replication and Load Sharing Approach

Shilpa Gambhir[#1], Er. Sonia Goyal[*2]

[#1] Student (Mtech ECE), Electronics and Communication Department, Punjabi University, Punjab, India
[*2]Assistant professor, Electronics and Communication Department, Punjabi University Patiala, Punjab, India

Abstract- **This research solves the problem of maximizing reliability of heterogeneous distributed computing system where random node can fail permanently. The reliability of the system can be achieved by executing all the tasks queued on its node before they all fail. As the mobile network is defined as a dense 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[1]. A chance of errors in the mobile distributed network is very high. Although battery capacities keep increasing, the execution demands of modern mobile devices continue to outstrip their battery lives. As a result, battery life is bound to remain a key factor in the design of mobile applications. In this paper we focused to improve the fault tolerance of the mobile distributed system so that it will take less time and energy consumption for execution when a node fails as compared to other systems.**

Keywords **- distributed system, reliability, mobile computing system, load sharing, candidate node.**

I. INTRODUCTION

The Distributed Computing System (DCS) is a collection of heterogeneous and geographically dispersed computing elements that cooperatively execute the applications. 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 distributed system is one in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing [2].DCS distributes intensive jobs into portions and send them to the machines for computations that are part of the distributed system. Nodes that are part of the distributed system execute their portion of the job and submit the results to job submission node [3].

In order to process the application, DCS firstly, divide them into set of tasks. Then it concurrently processes to these tasks on different processors of DCS. The performances of the DCS highly depend upon the assigning of tasks onto the processors. In literature, the technique of assigning tasks to the processors is known as task allocation. Task allocation highly affects the performance of the DCS. If the allocation of tasks isn't done carefully, processors of the system spend more time to transfer the tasks rather than performing useful computations. In literature this problem is called Thrashing.Distributed Computing System is heterogeneous in nature. So different type of hardware and software arerequired are required to build the distributed system.

Mobile computing system (MCS):Mobile Computing System is a distributed system where one of the processes is known as Mobile Node.Recent advances in wireless communication technology and portable computing devices have enabled rapid development of mobile computing systems. MCS is an example of DCS. In mobile computing systems, mobile nodes are equipped with wireless interfaces, and they remain connected to the network through wireless links even when they are mobile. There are several network models of mobile computing systems, including wireless Local Area Networks (wireless LANs), mobile ad-hoc networks (MANET) and cellular networks. Local Area Networks (LANs) are used to interconnect computers in a relatively small area using wires or cables. In wireless LANs, computers or mobile nodes communicate by means of high-frequency radio waves. There are several wireless LAN standards, among which, IEEE 802.11 (including 802.11, 802.11a, 802.11b, and 802.11g) is widely used.

ISSN: 2278 – 909X

A mobile computing system MCS={N, C} is composed of a set of nodes N and a set of channels C. The set of nodes N can be divided into two types, M={MH1, MH2,...,MHn} is the set of mobile nodes, which are able to move while retaining their network connections and S={MSS1, MSS2,..., MSSm} is the set of static nodes where MSS is the mobile supporting station with extra processing power and storage capabilities.
A cell is a geographical area covered by a MSS. A MH residing in a cell can directly communicate with MSS of the cell through a wireless channel. And a MH can reside in the cell of only one MSS at any time. MSS acts as head node and the MH acts as sub nodes. The computing between head node and sub nodes gets affected if any node fails. Because of mobility, an active MH can move from cell to another. Thus, when a MH moves from one cell to another, it gets disconnected from its MSS and it is not able to perform the task assigned to it. This is known as failure [4].
Types of failures that can occur in mobile computing system are:
i) Faulty MH, disconnection of MH due to low battery or overloading or movement of MH. This type of failure is called a node failure.
ii) Failure of static node ie MSS. This type of failure is called host failure.
iii) Wireless link failure known as communication link failure.

In this paper we deal with the case of node failure where a mobile node (MH)moves from one cell to other and gets disconnected from its MSS.

Fault Tolerance in Distributed System:
Fault tolerance is the ability of a system to withstand failure and continue to provide service in the event of an internal or external error. Fault tolerant systems are designed to ensure that in the event of a failure, crash, or a major user error, data are not lost and the system can continue to provide its specified services, thereby increasing the reliability and dependability of a system usually by masking the software or hardware faults[5].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. There are various fault tolerant techniques. Once the fault has occurred in a network, the next step is the reallocation of tasks among the working nodes. This is done using task allocation model.

II. RELATED WORK

Improving reliability of DCS has been an active area of research and interest for a long time. Even after so much work, task allocation after failure has been a challenge to meet the requirements of the systems. Many fault tolerant techniques and task allocation algorithmshave been proposed and adopted.

SajjadHaider, NaveedRiazAnsari ,Muhammad Akbar , Mohammad RazaPerwez and KhawajaMoyeezUllahGhori studied fault tolerance in distributed paradigms. They presented a comprehensive classification of errors, failures and faults that can be encountered in a Distributed environment [3].

Tome Dimovski, PeceMitrevski proposed a distributed transaction processing model in mobile environment which considers two communication scenarios, i.e. one when mobile hosts can connect to the fixed network and the second when they cannot. A decision algorithm is responsible for making a decision for a mobile host when it is disconnected from the fixed network for a certain period of time [6].

Jorge E. Pezoa, SagarDhakal, and Majeed M. Hayat studied the performance of DCS by redundancy approach. They present a framework to analytically characterize the service reliability of DCS in case of communication uncertainties and topological changes due to node deletions. The presented analysis is based upon the regeneration theory that exploited to derive a system of difference-differential equations characterizing the service reliability [7].

I. Maatouk, E. Chatelet, and N. Chebbo show the reliability of multi-state system with load sharing approach. They have presented an approach for evaluating the dynamic performance distribution of multi-states distributed computing system with dependent components. The dependency introduced the common cause failure and the load sharing between system components [8].

Vinod Kumar Yadav, MahendraPratapYadavand Dharmendra Kumar Yadav in [9] , tried to solve the problem of maximizing reliability of heterogeneous distributed computing system where random node

ISSN: 2278 – 909X

can fail permanently. They determined 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.

III.PROBLEM STATEMENT AND SOLUTION

Consider a DCS here with 'N' heterogeneous computing nodes. We also suppose that workload is divided into 'M' tasks. 'M' tasks are divided among 'N' nodes depending on various parameters. When a node fails before executing the task assigned to it, the task should be reallocated among other working nodes in the system with the task allocation model so as to avoid any data loss.
So the problem of re allocating the task to the working nodes arises.

Task allocation model: We have two types of task allocation models as follows:

1. Load balancing algorithm is based on the redistribution of processes among the processors during execution time. This redistribution is performed by transferring tasks from the heavily loaded processors to the lightly loaded processors with the aim of improving the performance of the application The major disadvantage of dynamic load balancing schemes is the run-time overhead due to: the load information transfer among processors, the decision-making process for the selection of processes and processors for job transfers, and the communication delays due to task relocation itself [9].

In this technique, the head node waits for some time after allocating tasks to each node. The node which doesn't reply in threshold time is said to have failed. And then the task is assigned to next appropriate working node. This technique is more time consuming as it has to wait for threshold time and assign task again to working nodes. So we have to reduce this time consumption because more time consumed will result in more battery consumption.

2. Load sharing algorithm improves the performance of the DCS in terms of network congestion, average response time, total sum of communication and service time. When a processor fails before executing the task assigned to it, the head node checks to the reliabilities of sub nodes and transfers the load to the working candidate nodes.

IV. PROPOSED TECHNIQUE

In proposed technique, we replicate the task to be assigned to each task and use load sharing algorithm where each node shares task on the basis of execution time and failure rate of each node. Due to this reason, we have named this task allocation approach as 'Replication and Load Sharing Approach'. Each node has some weight. Higher weight represents the node can handle more traffic. Load sharing technique is used to assign tasks to candidate nodes. But in this technique the whole task is assigned to all the nodes which store it in their respective buffers or memory. Each candidate node is given an instruction to perform particular part of the task. By doing so it takes less time to allocate its task to nodes again once a failure has occurred because the working nodes already have similar task in their storage. We use 'pinging and timeout' mechanism to detect node failure and then allocate the task to the node with highest weight.

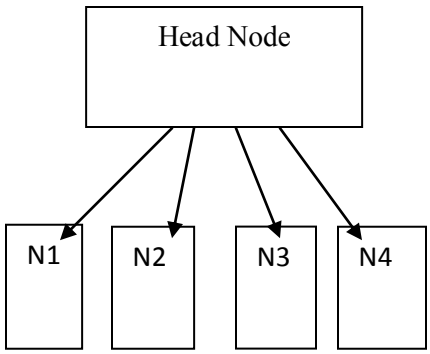

Figure1. Mobile computing system model

Figure 1 shows a mobile computing system with 1 head node and 4 sub nodes. Node 1, node 2, node 3, and node 4 with weights2, 5, 6, 8 respectively.
Maximum failure rate is 8 and maximum execution time is 10s.

- Failure rate of node 1 is 5 and execution time is 6s.
- Failure rate of node 2 is 9 and execution time is 5s.
- Failure rate of node 3 is 2 and execution time is 9s.
- Failure rate of node 4 is 7 and execution time is 7s.

The node whose failure rate is less than max failure rate and execution rate is less than max execution rate is selected as candidate node. Now in our case node 1, node 3 and node 4 are selected as candidate nodes. Suppose 'A+B+C' task is given to the head node

ISSN: 2278 – 909X

which divides it to the sub nodes. The task will be assigned to the all the nodes with the instruction that which part of the current task are to be performed by that particular node. Now node 1, node 3 and node 5 all have 'A+B+C' task.

Node 1 has to execute task 'A'and store task 'B' and task 'C'. Node 3 has to perform task 'B' and storetask 'A' and task 'C'. Node 4has to perform task 'C' and store task 'A' and task 'B'. Now for the case of failure, node 3 moves from its original place and fault occur before execution of task. The task which was assigned to node 3 i.e. task 'B' is to be allocated to working candidate nodes. Now the working nodes are node1 and node 4. The task 'B' is assigned to node 4 because its weight is higher than weight of node 1. It means node 4 can handle more traffic. In this way the task of the failed node i.e. task 'B' is assigned to node 4.
In the presented method, the time consumed for execution of the whole task is less and therefore the energy consumption is less.

The proposed idea has been simulated in MATLAB. The algorithm contains three major parts. The first part takes the 'M' tasks and 'N' processors as input and determines the candidate nodes on the basis of requirements of tasks and resources available on the processors. In the second part assignment of the tasks among most appropriate candidate nodes is done. In this assignment it considered the parameters: failure rate and execution time. In part three it handles the case of node failure. When node failed before executing all the tasks assigned onto it. It transfers the remaining tasks among the next appropriate candidate nodes. The next appropriate node is decided by considering the weight of the node. The node with higher weight is chosen.

The Figure 2 shows the comparison of processing time of the old method and new proposed fault tolerance method.

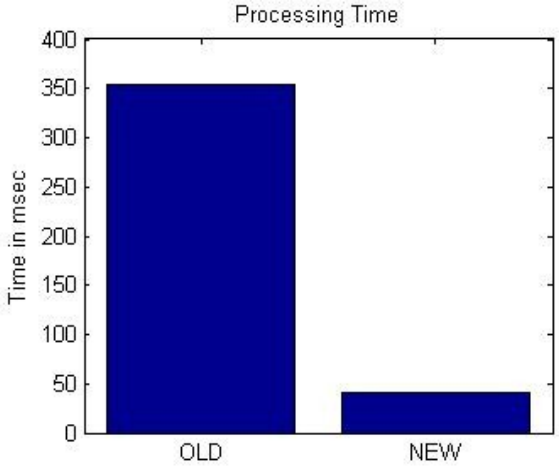

Figure 2

Energy consumed by the system in effectively distributing the task in case of failure is calculated by E*T, where E is energy consumed by the system in one sec and T is total time.

If the energy consumed by the system is 1.5 joules per second then the energy consumed is 1.5 * total time consumed. Thus the energy consumption Figure 3 has been plotted which shows the comparison of energy consumed by the old method and new proposed method.

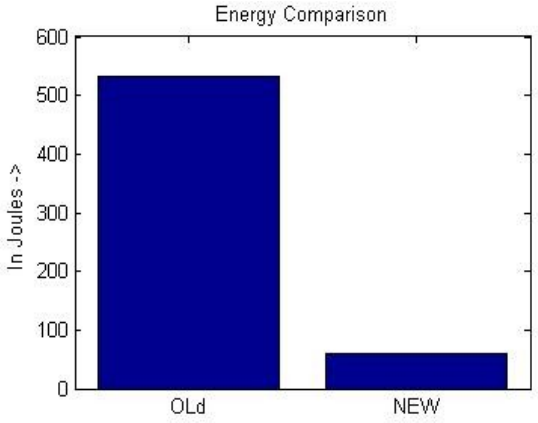

Figure 3

V. CONCLUSION

In this paper, we presented a technique that uses replication and load sharing for task allocation. Experimental results show that proposed method is far better than existing method as it takes lesser time to complete the task after node failure. Thus the proposed method improves the system reliability in terms of less time and energy consumption. Another advantage of our approach is that optimization is

ISSN: 2278 – 909X

fault-tolerant when the mobile node gets disconnected; the optimized application seamlessly switches to the other nodes in the network. We have evaluated our approach which optimized to use less energy and run faster, without compromising the fault-tolerance.

REFERENCES

[1] Prof. Elizabeth White, "Introduction to Distributed system", Distributed Software Systems.

[2] Petru Eles, "DISTRIBUTED SYSTEMS" http://www.ida.liu.se/~petel

[3] Sajjad Haider, Naveed Riaz Ansari, Muhammad Akbar, Mohammad Raza Perwez, Khawaja MoyeezUllah Ghori1, "Fault Tolerance in Distributed Paradigms", International Conference on Computer Communication and Management, 2011.

[4]Chaoguang Men, Zhenpeng Xu ,''Performance Analysis of Rollback Recovery Schemes for the Mobile Computing Environment'',IEEE International Conference on Internet Computing in Science and Engineering,2008.

[5] A. Avizienis, "The N-version Approach to Fault-Tolerant Software", IEEE Transactions on Software Engineering.

[6] 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

[7] Jorge E. Pezoa, Sagar Dhakal, and Majeed M. Hayat, Maximizing Service Reliability in Distributed Computing Systems with Random Node Failures: Theory and Implementation, IEEE trans. on parallel and distributed system, 2010.

[8] I.Maatouk, E.Chateet, N.Chebbo, Reliability of multi-states system with load sharing and propagation failure dependence, 978-1-4577-1232- 6/11/2011 IEEE.

[9] 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

[10] Rajwinder Singh and Mayank Dave, Senior Member, "Antecedence Graph Approach to Checkpointing for Fault Tolerance in Mobile Agent Systems", IEEE transactions on computing, 2013

[11] Shan Zhang and Jian-ping Wu, "Construction of Distributed and Heterogeneous Data Sharing Platform" ,International Conference on Web Information Systems and Mining, 2009

[12] 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

[13] 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

[14] Jinho Ahn, "Lightweight Fault-tolerance Mechanism for Distributed Mobile Agent-based Monitoring" IEEE, 2008

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

[16]Yong Li , Pan Hui, Depeng Jin , Li Su, Lieguang Zeng, "An Optimal Distributed Malware Defense System for Mobile Networks with Heterogeneous Devices'', 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks,2011

ISSN: 2278 – 909X