Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 8, August 2015, pg.330 – 334

REVIEW ARTICLE

To Improve Fault Tolerance in Distributed Computing System- A Review

Harjot Kaur[1], Raju Sharma[2]

Student[1], Assistant Professor[2], Department of ECE

BBSBEC, Fatehgarh Sahib

Abstract: A distributed computing is software systems in which components are located on different attached computers communicate and organize their actions by transferring messages. There are some challenges in distributed computing system. In this paper, we focus on fault tolerance which is responsible for the degradation of the system. A novel technique is proposed based upon reliability to overcome fault tolerance problem and re-allocate the task.

Keywords: DCS, reliability, master node time, execution time

1. Introduction

Computing System is a system which is a combination of number of computers and associated software sharing common memory. It can be peer-to-peer and multi-hop also. A distributed system is a layout in which multiple components that are on multiple computers but run as a single coherent computer system. A distributed system linked by local networks and interlinked with each other's physically [1]. 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. Distributed Computing System is heterogeneous in nature. Due to it varied nature there is a need of various hardware's and software's of different kind. Distributed system is better than centralized system in various ways. First point is it does not have central controller. Therefore there is no chance of system failure if central system collapses. In distributed system computers are located at faraway places. They are connected with number of servers. If there is chance of any server failures, it gets data from other servers also. Another point is scalability. Due to scalability nature, a number of computers can be added at any time to increase its strength. Another point is redundancy. As we discussed it is combination of various smaller machines it is not that much expensive. It can easily affordable and extendable. Distributed computing utilizes a network of many computers which accomplishing a portion of the entire task. A distributed program is a computer program that runs on distributed system. A distributed programming is the process of writing such types of languages [4]. Grid computing and Cluster computing are types of distributed computing systems.

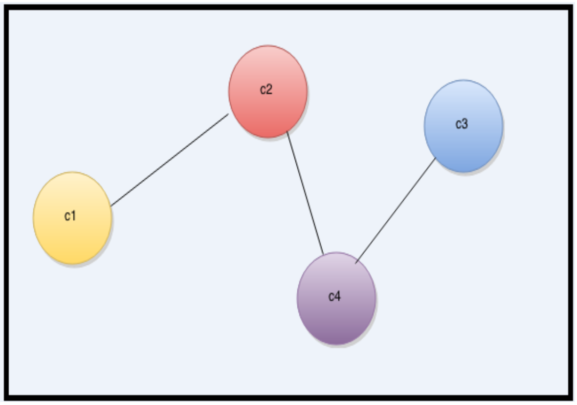

Fig 1: Distributed system

A distributed system has a number of independent computers allied up 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, incorporated computing facility [5].

1.1 Mobile Computing System: Mobile computing system is also one of the types of distributed computing system. In this network nodes are called mobile node. It is example of the DCS. In this type of system, mobiles nodes have their own wireless interface [5]. They are in linked with the interface through wireless link even when they are mobile. Mobile computing system has various models in the field of networking. It is used in wireless LAN, MANET, WAN, MAN. In all the fields there is a requirement of wireless device to interlink the entire interface together. In mobile computing system we need a node and channel for the communication.

1.3 Issues in Distributed Computing System: There are some issues in distributed system which is responsible to lower down its rate. These issues are:

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. Fault Tolerance: The system must be resistance to faults. In future if any fault may occur it doesn't degrade its performance. Fault can be occur due to mobility, overloading, load imbalance and many more factors.

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.

2. Review of Literature

We have studied various papers to improve fault tolerance of the system. In this section we will be discussed various techniques and algorithms for improving performance of the system.

In paper [1] they proposed a model in mobile computing which have two scenarios. They considered two cases one case is when mobile hosts connect with the fixed network. Second case is when mobile host does not connect with the host. It has one decision tree algorithm which decides when node has to connect with the fixed network and when node has to disconnect.

In paper [2] they mentioned a brief categorization of errors, faults and failures that are encountered in a distributed environment.

In paper [3] they presented a technique to remove the problem occurred due to failure of permanent node. Basically they tried to remove overcome the complexity which is occurred due mobility of the node. They proposed a load sharing technique to maintain the performance of the system.

In paper [4] they proposed an algorithm which is based upon the checkpoint technique. It is used to make the system fault free and improve their performance based upon the antecedence graphs. They also proposed a future work in which they mentioned that integrating graph and non-graph based scheme to get high fault tolerance system.

In paper [5] they proposed a solution for the dynamic allocation technique. In this they proposed a technique in which each process may execute task per process. Whenever it shifted to the next phase reallocation cost is added to the process. At this time it may be at the same processor or may shift to another processor. By doing so, at the end we will get an optimal cost at the optimal cost.

In paper [6] they proposed a technique which is based upon some provisos like its input should not be zero and it is not available for all the users. 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 [7] they represented Mobile Agent technology to improve the flexibility and doing by promise as a powerful agent and its mechanism. Mobile agent systems must also provide a 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. 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 paper [8] they presented a 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 paper [9] presented mobile agent based fault prevention and detection technique where the team of mobile agents monitor each host in mobile agent based system. 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.

3. Fault Tolerance in Distributed Computing Systems

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. 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 [11]. 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. Fault Tolerance is important method in grid computing because grids are distributed geographically in this system under different geographically domains throughout the web wide. The most difficult task in grid computing is design of fault tolerant is to verify that all its reliability requirements are meet [12].

3.1 Techniques for Fault Tolerance: The best is considering that system which is free from all the faults and has immaculate and impeccable performance. So there are various techniques which premeditated to make the system fault tolerant. These techniques are:

3.1.1 Hardware Resilience: Hardware Resilience is the technique which is related to the hardware transparency. In this technique unit hardware transparency is used to handle the reliability of the network. It is memory error correction technique [6].

3.1.2 Application based Resilience: It deals with faults using information about the application used by it and makes the developer more intelligent.

3.1.3 Intercrosses Communication: In distributed system to share information process must be communicated to one another. For this there is a need of synchronization between all the processes. But it should be in controlled manner so that there process cannot judge the speed of another process. All the entire scenario communication is done by the message passing scheme. There is no need of shared memory for message passing [8].

a) Synchronization Message Passing: This process is without buffering. Send command is on hold until receive command has not clear or executed. It has one synchronization point on which both sender and receiver has synchronized them with that point. Assertions are made in this process.

b) A synchronization Message Passing: This process takes communication and synchronization both processes separately [9].

3.1.4 Replications: Replication means to make multiple copies of similar data on the servers**.** To make any action successful replication is needed. Suppose if one node fails and there is no replication of that data it affects the performance of the system. But there are replications of data than user may get data from other servers in case of any node failure also. It is proxy based monitoring technique which is applied in distributed systems. It has two strategies either Active or Passive. It helps to remove overhead issues and complexity. It is well known technique which is used to enhance the availability. But replications may produce some serious problems like inconsistency [3].

Replications help to improve performance of the system. By using numerous protocol users receive up-to-date data. It also has some limitations like it is expensive. It also helps to boost availability of the system because of multiple copies. But it requires update data every time. Moreover it helps to make system fault tolerant [5].

3.1.5: Load Balancing Algorithm: Load balancing algorithm is based on the reallocation of processes during execution time among the processors. The main aim of the system is to improve performance of the system. This can be done by allocating task to the light weighted task from heavy weighted task. Run-time overhead is the disadvantage of dynamic load balancing schemes 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].

***3.1.6: Check pointing*:** Basically this technique is used to restore the process to certain point after failure occurs. Fault Tolerance can be achieved through various types of redundancy. Check-point start is the common method. In this method an application starts from the earlier checkpoint after a fault. Application may not be able to meet strict timing targets.

4. Proposed Methodology

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 which helps to reduce fault tolerance of the system and increase performance of the system.

In this work, new formula of reliability is added which will calculate reliability of each node which are responsible for the task execution. All available nodes have reliability value 1 in the starting phase of the project. The formula is applied which based on the maximum failure rate and maximum execution time. Each node in the network has its own failure rate and execution time. On the basis of these two and number of tasks which are going to be executed, reliability value is calculated. The node which has maximum reliability value will execute the task on the node which changed its position. The formula for reliability calculation is given below

1. AB=Maximum execution time+ Maximum Failure rate
2. BC=execution time of each node + failure rate of each node
3. DE=BC*number of task for execution
4. Reliability= AB-DE

5. Conclusion

Distributed Computing System is heterogeneous in nature. Due to it varied nature there is a need of various hardware's and software's of different kind. Mobile computing is one of the types of distributed computing. There are various issues of distributed computing like scalability, availability and fault tolerance. In this paper, various techniques of fault tolerance in distributed computing system have been studied. Every technique has its pros as well as cons also. In this paper a new technique has been proposed to improve accuracy of the system. During the case of node mobility and node failure it can be used to improve performance and makes system fault tolerant.

References

[1] 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

[2] 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

[3] 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

[4] 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

[5] Dr. Kapil Govil, "A Smart Algorithm for Dynamic Task Allocation for Distributed Processing Environment" International Journal of Computer Applications (0975 – 8887) Volume 28– No.2, August 2011

[6] 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

[7] 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

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

[9] 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

[10] 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.

[11] Alexandru Costan, Ciprian Dobre, Florin Pop, Catalin Leordeanu, Valentin Cristea, "A Fault Tolerance Approach for Distributed Systems Using Monitoring Based Replication", IEEE, 2010

[12] 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