© 2018 IJSRST | Volume 4 | Issue 11 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology
DOI : https://doi.org/10.32628/

Improvement of Distributed Computing Performance with Fault Tolerant Algorithms

Bandru Sanjeev , Vattipally Vinod Reddy
Assistant Professor, IT Department, Sreednidhi Institute Science and Technology, Hyderabad, Telangana, India

ABSTRACT

Computer science has many of it applications one of it is Distributed Computing (DC) which tells about the distributed systems. The working of distributed systems is exchanging information between different networks located in various places to share the problem. Computing technologies performing well in the information services still we have few problems related to DC among which fault tolerance is one which impacts the entire system performance. In this paper we tried to provide solution to this problem with the concepts of reliability and availability of resources to reallocate the job.

Keywords : Distributed Computing, Concurrency, SOA, Reliability, Execution and Master Node Time.

I. INTRODUCTION

Distributed systems consists of group of autonomous computer systems brought together to provide a set of complex functionalities or services. The computer systems are geographically distributed and are heterogeneous in nature. Distributed systems appear as one local machine to the users. These systems are advantageous as they provide scalability of software and resources dynamically.

Distributed systems are required to be dependable having following characteristics.

- Systems must be available, must not fail.
- Must fulfil timing and requirements.
- Systems output is required to be accurate.
- System must be secure
- System must provide safe mode operations

Thus, the dependability refers to reliability, availability, survivability and safety. To achieve these system characteristics, system must be designed to have the ability to handle faults and failures dynamically.

Fault tolerance is the dynamic method that's used to keep the interconnected systems together, sustain reliability, and availability in distributed systems. The hardware and software redundancy methods are the known techniques of fault tolerance in distributed system. The hardware methods ensure the addition of some hardware components such as CPUs, communication links, memory, and I/O devices while in the software fault tolerance method, specific programs are included to deal with faults. Efficient fault tolerance mechanism helps in detecting of faults and if possible recovers from it.

II. METHODS AND MATERIAL

A. Distributed System

Distributed system are systems that don't share memory or clock, in distributed systems nodes connect and relay information by exchanging the information over a communication medium. The different computer in distributed system has their own memory and OS, local resources are owned by the node using the resources. While the resources that is been accessed over the network or communication medium is known to be remote resources. Figure 1 shows the communication network between systems in the distributed environment. In distributed system, pool of rules is executed to synchronise the actions of various or different processes on a communication network, thereby forming a distinct set of related tasks.

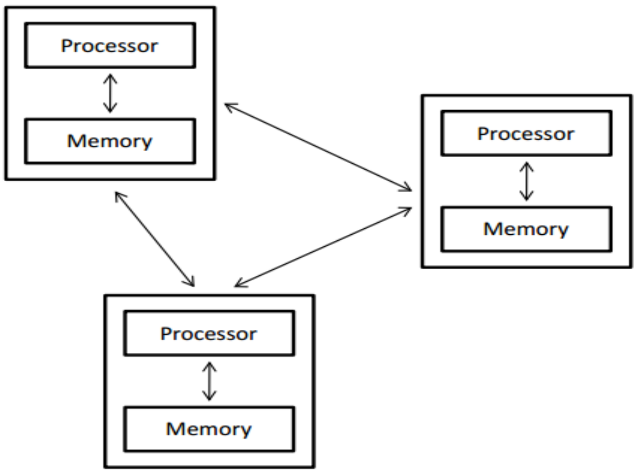

Figure 1. Distributed system

B. Distributed System Architecture

The architecture of distributed system is built on existing OS and network software. Distributed system encompasses the collection of self-sufficient computers that are linked via a computer network and distribution middleware. The distribution middleware in distributed system, enables the corresponding computers to manage and share the resources of the corresponding system, thus making the computer users to see the system as a single combined computing infrastructure. Middleware is the link that joins distributed applications across different geographical locations, different computing hardware, network technologies, operating systems, and programming languages. The middleware delivers standard services such as naming, concurrency control, event distribution, security, authorization etc. Figure 2 shows the distributed system architecture, with the middleware offering its services to the connected systems in the distributed environment.

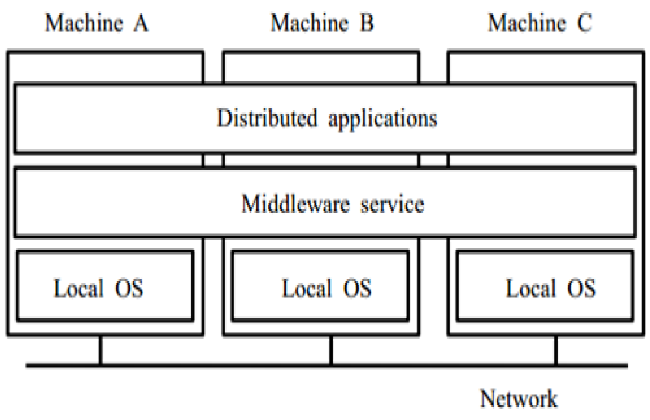

Figure 1. Simple DC architecture

C. Fault Tolerance Systems

Fault tolerance system is a vital issue in distributed computing; it keeps the system in a working condition in subject to failure. The most important point of it is to keep the system functioning even if any of its part goes off or faulty.

Availability: This is when a system is in a ready state, and is ready to deliver its functions to its corresponding users. Highly available systems works at a given instant in time.

Reliability: This is the ability for a computer system run continuously without a failure. Unlike availability, reliability is defined in a time interval instead of an instant in time. A highly reliably system, works constantly in a long period of time without interruption.

Safety: This is when a system fails to carry out its corresponding processes correctly and its operations are incorrect, but no shattering event happens.

Maintainability: A highly maintainability system can also show a great measurement of accessibility, especially if the corresponding failures can be noticed and fixed mechanically.

D. Dynamic and Self Adaptive Fault Tolerance Algorithms

Current trend of distributed applications demand dynamic and self adaptive fault tolerance techniques. The techniques must be capable of handling frequent and multiple faults at the run time and also adaptive to different run time conditions. Adaptive programming model can be used to develop such techniques [5]. There is lot of research scope for developing programming models for implementing adaptive techniques. In large scale distributed system failure detection is fundamental task for ensuring fault tolerance. Failure detectors must be capable of working asynchronously and independent of application flow. The major issue with these is their ability to scale for large number of nodes. Handling multiple faults is becoming crucial as number of nodes scale in distributed system. Single point failure often cause serious problems, hence must given much attention.

III. LITERATURE SURVEY

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. Choosing a fault tolerance technique appropriate for a system is an important task [6]. We can evaluate the techniques based on some major factors like consistency management, multiple faults handling, and efficiency of working procedure, multiple failure detection and performance. Comparison shows these techniques are reliable and have the capability of detecting and handling multiple faults. The performance can be improved by working on the critical issues.

Table -1: Comparison of fault tolerance technique

Factors	Replication based	Process level redundancy	Check point based	Fusion based
Consistency management	Strategies like active or passive replication	Can be easily implemented	Can be handled by avoiding orphan messages	Has to be implemented among back up machines
Multiple faults handling	Affected by degree of replication	Affected by number of redundancy processes	Affected by check point processes	Affected by number of back up machines
Working	Requests are forwarded to replication	Redundant process	Snapshots saved on stable storage for recovery	back up machines
Multiple failure detection	Accurate and adaptive	Reliable, accurate and adaptive	Reliable, accurate and adaptive	Reliable, accurate and adaptive

IV. RESULTS AND DISCUSSION

A. Fault Tolerance In Distributed Computing

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 [1]. 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 [2].

B. 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. Runtime 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 [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.

C. Proposed Model

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.

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

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.

V. CONCLUSION

Since distributed computing comprises of various software's and hardware's hence exhibits heterogeneity in nature and leads issues like scalability, availability and fault tolerance. We used various techniques to avoid fault tolerance in DC for better performance. Accuracy of the system was calculated by using node mobility and node failure and finally rate of fault tolerance. The fault tolerance approaches discussed in this paper are reliable techniques. Further the performance of these can be improved towards achieving high reliability. There is lot of research scope in minimizing recovery time of existing techniques and implementing dynamic adaptable techniques. We left the issue of integration in distributed computing as future work.

VI. 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]. 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.

[3]. 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

[4]. 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.

[5]. Ravindra Changala, "Retrieval of Valid Information from Clustered and Distributed Databases" in Journal of innovations in computer science and engineering (JICSE), Volume 6, Issue 1,Pages 21-25, September 2016.ISSN: 2455-3506.

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

Authors:

Mr. B. Sanjeev completed his M.Tech from JNTUH Hyderabad. His area of interests are distributed computing and cloud computing. He has a couple of publications in the mentioned domains. He is presently working as an Assistant Professor in the CSE Department at Sreenidhi Institute of Science and Technology, Hyderabad.

Mr. V Vinod Reddy completed his M.Tech from JNTUH Hyderabad. His area of interests is distributed computing and Soft Computing. He has a couple of publications in the mentioned domains. He is presently working as an Assistant Professor in the CSE Department at Sreenidhi Institute of Science and Technology, Hyderabad.

Cite this article as :

Bandru Snajeev, Vattipally Vinod Reddy, "Improvement of Distributed Computing Performance with Fault Tolerant Algorithms", International Journal of Scientific Research in Science and Technology (IJSRST), Print ISSN : 2395-6011, Online ISSN : 2395-602X, Volume 4, Issue 11, pp.293-299, November-December-2018.
Journal URL : http://ijsrst.com/IJSRST18401153