

A System Model of Fault Tolerance Technique in the Distributed and Scalable System: A Review

Deepika Dhawan¹, Dr Faiyaz Ahmad², and Manish Madhava Tripathi³

¹PG Scholar, Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, India

²Assistant Professor, Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, India

³Associate Professor, Department of Computer Science & Engineering, Integral University, Lucknow, Uttar Pradesh, India

Correspondence should be addressed to Deepika Dhawan; maildeepika21@gmail.com

Copyright © 2022 Made Deepika Dhawan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT- Fault tolerance is one of the most crucial concerns in distributed systems. Fault tolerance system is very difficult to implement due to its dynamic nature and complex services. Several research efforts consistently being made to implement that tolerance in a distributed system. Some recent surveys try to incorporate the several fault tolerance architectures and methodologies proposed for a distributed system. This paper gives a systematic and comprehensive interpretation of different fault types, their causes, and various fault-tolerance approaches used in a distributed system. The paper presents a broad survey of various fault tolerance frameworks in the context of their basic approaches, fault applicability, and other key features. We investigate the different techniques of fault tolerance which is used in a distributed and scalable system. Scalability is an important factor in distributed systems. It describes the ability of the system to dynamically adjust its own computing performance by changing available computing resources and scheduling methods. The focus of this paper is on types of faults occurring in the system and fault detection techniques. A fault can occur in the system due to the link failure or for any other reason. An appropriate fault detection technique can avoid a loss and save from system failure. The main objective of the fault-tolerant computer system is to continue operating uninterrupted despite the failure of one or more of its components. In the early days computer systems were not distributed and they also did not share resources. Now, most of the computers are distributed. They work independently on a common task. So, if one system gets any fault then the other systems will take over the computation of the fault system. The user will not get any issues with his tasks.

KEYWORDS- Load balancing, Fault tolerance system, Server authentication.

I. INTRODUCTION

Distributed systems are made up of several components, developing a system that is hundred percent fault-tolerant is practically very challenging. Every system has its own memory and resources, and they share common peripheral devices. There are three kinds of problem that occurs in distributed systems: -

- Failures
- Errors
- Faults

All these are interrelated. Fault is the root cause where the problem starts, after the fault occurs then we will notice the errors in the system working. Failure is the state in which the system fails to give an outcome i.e. failure is the final outcome.

Two main reasons for the occurrence of a fault:

- Node failure
- Malicious Error

Fault-tolerant systems use backup components that automatically take the place of failed components, these include Hardware systems, Software systems, and power resources. In a large-scale environment, Fault tolerance is common to a network system. Basically Fault tolerance refers to the ability of a system to continue operating without interruption when one or more of its components fail. The main objective of creating a fault-tolerant system is to prevent disruption due to a single point of failure. Fault-tolerant systems use backup components that automatically take place of failed components, ensuring no loss of service. These include:

Hardware Fault tolerance: It is the ability of a system to execute the required function in the presence of one or more hardware faults. There are two approaches namely BIST i.e. Built In Self-test, and TMR i.e. Triple Modular Redundancy. In BIST technique system carries out the test of itself after a certain period again and again. When the system detects a fault, it switches out the faulty component. Basically, the system reconfigures itself in case of fault.

In triple Modular Redundancy, three redundant copies of critical components are generated, and all three copies are run concurrently. Voting of the result of all redundant copies are done and majority result is selected. It can tolerate the occurrence of a single fault at a time.

Software Fault tolerance: This is the ability of the software to continue its normal operation despite the presence failures. There are different mechanisms for software fault tolerance i.e.

- Recovery blocks
- N-version Software
- Self-checking Software

Power Sources: If the primary sources fail then the other alternative approaches are used to support them i.e. to use power generators that ensure storage and hardware, heating, ventilation, and air conditioning (HVAC) continue to operate

as normal. One approach is to run devices on an uninterruptible power supply (UPS).

II. REVIEW LITERATURE

Mohammed Amoon presented a Fault-Tolerant Scheduling System Based on Checkpointing for Computational Grids says that job checkpointing is the most common technique for fault tolerance. The efficiency of checkpointing is depended on the selection of the checkpoint interval. If the inappropriate checkpointing interval is selected, then the job execution can be delayed. In this paper, a fault-tolerant job scheduling system based on checkpointing technique is evaluated. The system uses the failure rate of the resources to calculate the checkpoint interval for each job[1].

Dr. J Meena Kumari and Shaima'a, Ghamdan presented A Study on Fault Tolerance Solution say that Nowadays cloud computing has gained popularity . Node failure is the major concern when applications run in a server. This is called a fault tolerance. During the execution of application reliability and availability is the main issue which may otherwise affect the quality of service provided by the cloud service providers. This paper addresses with the various fault tolerance techniques used in the cloud environment such as fault masking , check pointing, Job Migration, Task Resubmission and Replication. Replication is the most common method used in fault tolerance. This paper highlights replication fault tolerance technique [2].

L. B. Bhajantri and N. Nalini, “A fault tolerance approach to topology control in distributed sensor networks,” is a vital issue in distributed computing. Distributed system uses different fault tolerance mechanism to prevent multiple system failures on multiple failure points.This paper highlights mechanisms such as replication , high redundancy and high availability of the distributed services [3,8].

S. Dave and A. Raghuvanshi, “Performance improvement in Distributed Systems through Replication and Checkpointing, for Workflow Applications on Computational Grids say that now a days complex scientific workflows are executed on computational grids. Due to the unreliable nature of large scale grid infrastructure reliability issue arises. To address the reliability challenges fault tolerance mechanism like over-provisioning and checkpoint recovery are used in grid application. This paper proposed new approaches that combine these fault tolerance techniques with existing workflow scheduling algorithms [4].

A. Munir, J. Antoon, and A. Gordon-Ross, “Modeling and analysis of fault detection and fault tolerance in wireless sensor networks that in real-time distributed system if fault is not detected and recovered in time it result into a system failure. The main focus of this paper is on fault detections and the recovery techniques used in real time distributed systems, There are different types of faults such as network fault, Physical faults, Media faults , processor faults, process faults and service expiry fault. An appropriate fault detector and fault tolerance technique is used in real time distributed systems [5].

Y. Lai *et al.*, “Identifying malicious nodes in wireless sensor networks based on correlation detection, “Reliable Task Allocation in Heterogeneous Distributed System with Random Node Failure: Load Sharing Approach, 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 [6].

Rajwinder Singh, Mayank Dave, “Using Host Criticalities for Fault Tolerance in Mobile Agent Systems 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 integrating graph and non-graph-based schemes to get a high fault tolerance system [7].

III. TYPES OF FAULT

There are different types of faults which can occur in distributed system. These can be classified as follows:

Table 1: Types of faults and correction technique

S. No.	Types of Faults	Method
1.	Network fault: fault occurs due to network partition, packet loss, packet corruption, destination failure, link failure, etc[1].	Reliability, Data corruption, and Automatic path migration
2.	Physical Faults: this fault can occur in hardware like fault in CPUS, fault in memory, fault in storage, etc[2].	Redundant array of independent disks
3.	Media Faults: fault occurs due to media head crashes.	Error detection and Error correction techniques
4.	Processor Faults: fault occurs in processor due to operating system crashes, etc[3].	Redundant execution, where computation is performed multiple times and the results are compared.
5.	Process Faults: fault occurs due to shortage of resource, software bugs, etc.	Recovery blocks N-version software Self-checking software

IV. DESCRIPTION OF LOAD BALANCING PROBLEM

Why the Load Balancing Problem Occur?

Load balancing strategy is also helpful in fault tolerance in distributed systems [8]. In distributed systems or in multiprocessor systems [8] , some of the nodes may be heavily loaded while others are lightly loaded[8] during the computing of the jobs [8]. This suggests the degradation of computing and degradation of the performance of the entire system . When a fault tolerance system is work with a load balancing system, this suggests that it is possible to improve

the overall performance of a multiprocessor system by transferring jobs from the heavily loaded nodes to the lightly loaded nodes [8]. This type of processing power-sharing is called load balancing, which can be classified into two categories: static and dynamic policies. A static policy uses only the information about the average behavior of the system to make job transfer decisions [8]. A policy that considers the current system state and reacts accordingly is called dynamic or adaptive policy [8]. A multiprocessor system is a new form of a parallel processing system [8]. The paper targets achieving reasonable task allocation in the system, making it operate at higher performance levels [8].

A. CLASS ATTRIBUTE OF WEBPAGES

Class attributes available on webpages is also playing significant role in fault tolerance and Load balancing in distributed systems. The main contribution of this system is followed with some steps [9]. First, we convert the HTML web page to the XHTML by this we can standardize the tags of the web page [9]. Then we segment it using the DOM tree method with the help of the DOM Inspector tool then we apply our class attribute approach through the classes we get by the DOM tree [9].

B. APPLICATIONS [9]

Information extraction from a noisy “junk” or unrelated content in web pages has plenty of applications [9]. It incorporates information retrieval and indexing to reformatting for better human consumption [9]. A user is good to differentiate between the related content and the unrelated content [9]. But with this semi-structured tool, the user can do it better and more efficiently [9]. But a typical automated agent would be implemented to process a web page in entirely with the noisy content mixed with relevant article content [9]. For information extraction [9], the Web is worth the trouble because it opens the possibility of achieving an elusive goal in Artificial Intelligence (AI), broad-coverage domain knowledge [9]. AI systems depend to a great extent on having knowledge about the domains in which they operate [9], and such knowledge is typically expensive to enter the system [9]. Furthermore, the knowledge must be entered for every different domain in which an application is to operate [9]. The Web contains knowledge about all kinds of different domains, but in a format that is not readily usable by AI systems [9]. WIE promises to bridge the gap between the Web and AI [9]. While a human user can intuitively separate the main content from the “junk” content, often users feel tired by having constantly track main content areas and focus away to unrelated “junk” [9]. Moreover, for viewing the web content on a smaller display it may be desirable to re-format the extracted main content article content and remove or reduce unrelated content for some applications [9].

C. Policies in Load balancing Algorithms [10]

Load balancing algorithms can be described by the implementation of the following policies Information strategy [10], which specify what workload information is to be composed and when it must be collected and from where [10], triggering strategy which determines the suitable period to begin a load balancing method [10]. Then the Resource type strategy classifies a resource to be a server or receiver of tasks according to the availability of status [10]. Location strategy uses the output of the resource type strategy to find

a suitable partner for a server or a receiver [10], Selection strategy defines the tasks that should be migrated from overloaded resources to idle resources [10].

D. Two Servers Password Authentication Two [11]

Two server confirmation instruments are painstaking to be protected for confirming a user in Internet grounded atmosphere [11]. As the quantity of amenities delivered operational is day by day snowballing [11], operators proposing to use numerous operational amenities are also cumulative [11].

Quantum Channel for two Server Password Authentication. In significant cryptography [11], significant key circulation etiquettes employment dramatic instrument to allocate meeting answers and community deliberations to checked for listeners and authenticate the accuracy of a conference significant [11].

E. Issue in Real-time Distributed System

In distributed System, Fault tolerance is the technique to give the required services in the presence of fault within the system. The main aim is to provide services as per requirement in the presence of faults.

In real time distributed system there are three main issues [5,6].

Feasibility – in real time distributed system feasibility means that a task should be finished on its deadline even though there is an error or fault within a system.

Reliability- this means availability of end-to-end services and the ability to experience failures, without impacting operations.

Scalability- it is the property of a system to handle growing amount of work by adding resources to the system.

Now the question is how these fault can be detected and removed from the system. In distributed system task should be finished on its deadline.

A real time task can be classified into either hard or soft real time task depending on the consequences of a task missing its deadline. In hard real time a task should be finished by its deadline sharply but in soft deadline task can finished nearby its deadline[8].

F. Failed System failure

After failure system can behave in three ways such as:

Fail Stop System: In this system immediately stops to sending any message and does not respond any message receiving on network. It results a permanent fault in the system and there is no output.

Byzantine System. In this system gives wrong output after failure [6,7].

Fail fast system : In this system does not perform any operation once it has failed.

G. Active and passive fault tolerance

Fault tolerance can be classified into passive and active, considering how it is achieved. A system with passive fault

tolerance does not react in any special way to compensate for the effect of faults, but by exploiting the intrinsic redundancy and fault masking, built into its structure, which efficiently masks the fault effects ensuring correct outputs despite faults [12]. The system is designed to mask, by compensation, a given maximum number of faults. No diagnostics, fault detection and localization, relearning, or reconfiguration are required. A system with active fault tolerance explicitly and dynamically recognizes and manages its redundant resources to compensate the fault effects (by adaptation, retraining or self-repairing mechanisms). It requires special detection/localization and supervising/control components,

whose design could be complex and intrusive [12]. In most cases, active fault tolerance makes it possible to recover from faults by reallocating the tasks performed by the faulty elements to the fault-free ones [12]. Generally, it is more difficult to achieve the same fault tolerance in an active approach as with a passive one, because not all the faulty scenarios can be considered at the system design, and no repair or reconfiguration is possible afterwards. In a hybrid approach, each approach can complement one to each other; a static base configuration masks a given number of faults, while faulty modules can be detected online and replaced within fault-free ones [12]

V. APPROACHES

Table 2: Multiple fault handling approach

S. No.	Type of Fault	Method	Factors			
			Working	Consistency Management	Multiple fault handling	Fault detection
1.	Replication	Replication is the process of creating multiple copies of data objects or processes	Request can be sent to one replica system to the other replica system	It is ensured by linearizability, sequential consistence and casual consistency	Affected by degree of replica	adaptive replicas creation algorithm used
2	Process level redundancy	This technique is used for faults that are short lived and less severe , this kind of fault is called transient faults.	Redundant processes	Can be easily implemented	Affected by the number of redundant processes	Reliable and adaptive
3	Checkpoint and Roll back based	In this technique current computation state is stored in a stable storage device and this is done occasionally. At the time of node failure, the operation is rolled back to most recent state.	In this two types of check point technique used ie coordinated and uncoordinated techniques.	Can be handled by avoiding orphan messages	Affected by check point scheduling	Reliable , accurate and adaptive
4.	Fusion based	The major problem of replication fault based technique is to manage backups. Because the backups increase as faults increase and the cost of management is very expensive, the fusion based technique solves that problem.	Requires fewer backup machines compared to the replication-based technique	Affected by number of back up machines	Backup machines	Reliable , accurate and adaptive

VI. CONCLUSION

Fault tolerance is a major part of distributed system. This research paper showed the different type of fault tolerance techniques in distributed system such as Replication based fault tolerance technique, Process level Redundancy technique, Checkpointing and Roll back technique and Fusion based technique. Choosing a fault tolerance technique appropriate for a system is an important task. It is clear that all the methods have capability to handle multiple faults All the fault tolerance showed in this paper are accurate, reliable, and adaptive techniques[Table 2].

REFERENCES

- [1] F. G. Khan, K. Qureshi and B. Nazir presented Performance evaluation of fault tolerance techniques in grid computing system says that fault tolerance techniques are important for the efficient utilization of resources. This paper presents a different parameters such as throughput, turnaround time , waiting time and network delay for the evaluation of the various fault tolerance techniques
- [2] A Fault Tolerant Scheduling System Based on Checkpointing for Computational Grids Mohammed Amoon Computer Science and Eng. Dept., Faculty of Elec. Eng., Menofia University, Egypt King Saud University, P. O. Box: 28095-11437 Riyadh-Saudi Arabia m_amoon74@yahoo.com
- [3] L. B. Bhajantri and N. Nalini, "A fault tolerance approach to topology control in distributed sensor networks," Proc. 2012 IEEE Int. Conf. Adv. Commun. Control Comput. Technol. ICACCCT 2012, no. 978, pp. 208–212, 2012, doi: 10.1109/ICACCCT.2012.6320772.
- [4] S. Dave and A. Raghuvanshi, "Performance improvement in Distributed Systems through Replication and Checkpointing," Int. J. Comput. Appl., vol. 42, no. 19, pp. 17–21, 2012, doi: 10.5120/5800-8039.
- [5] A. Munir, J. Antoon, and A. Gordon-Ross, "Modeling and analysis of fault detection and fault tolerance in wireless sensor networks," ACM Trans. Embed. Comput. Syst., vol. 14, no. 1, 2015, doi: 10.1145/2680538.
- [6] Y. Lai et al., "Identifying malicious nodes in wireless sensor networks based on correlation detection," Comput. Secur., vol. 113, 2022, doi: 10.1016/j.cose.2021.102540.
- [7] R. K. and A. Nanoti, "A Review on Genetic Algorithm for Fault Tolerance in Distributed Sensor Networks," Ijarce, vol. 5, no. 12, pp. 346–348, 2016, doi: 10.17148/ijarce.2016.51281.
- [8] Mohammad Haroon And Mohammad Husain, "Analysis Of A Dynamic Load Balancing In Multiprocessor System," International Journal of Computer Science Engineering and Information Technology Research (IJCEITR) ISSN 2249-6831 Vol. 3, Issue 1, Mar 2013, 143-148
- [9] S. Srivastava, M. Haroon, and A. Bajaj, "Web document information extraction using class attribute approach," Proc. - 4th IEEE Int. Conf. Comput. Commun. Technol. ICCCT 2013, pp. 17–22, 2013, doi: 10.1109/ICCCT.2013.6749596.
- [10] R. Khan, M. Haroon, and M. S. Husain, "Different technique of load balancing in distributed system: A review paper," Glob. Conf. Commun. Technol. GCCT 2015, no. Gcct, pp. 371–375, 2015, doi: 10.1109/GCCT.2015.7342686.
- [11] M. Salman Husain and D. M. Haroon, "A Review of Information Security from Consumer's Perspective Especially in Online Transactions," Int. J. Eng. Manag. Res., vol. 10, no. 4, pp. 11–14, 2020, doi: 10.31033/ijemr.10.4.2.
- [12] C. Torres-Huitzil and B. Girau, "Fault tolerance in neural networks: Neural design and hardware implementation," 2017 Int. Conf. Reconfigurable Comput. FPGAs, ReConFig 2017, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/reconfig.2017.8279793.