FAULT TOLERATION IN DISTRIBUTED SYSTEM

Disha Maini, Abhinav Shrestha

Department of Information Technology, Dronacharya College Of Engineering, Gurgaon

***Abstract-* Fault-tolerant computing is the art and science of building computing systems that continue to operate satisfactorily in the presence of faults. A fault-tolerant system may be able to tolerate one or more fault-types including -- i) transient, intermittent or permanent hardware faults, ii) software and hardware design errors, iii) operator errors, or iv) externally induced upsets or physical damage. In past there have been cases where critical applications buckled under faults because of insufficient level of fault tolerance. Various issues are examined during distributed system design and are properly addressed to achieve desired level of fault tolerance. In this research paper , we will focus on integrating security and fault-tolerance into one , general-purpose protocol for a secure distributed voting.**

I. INTRODUCTION

A faulty system attributable to any reason throughout process some task will causes some damages. A task running on real time distributed system ought to be feasible, reliable and ascendible. The real time distributed system like nuclear systems, robotics, traffic management systems, grid etc. are extremely dependable on deadline. A fault in real time distributed system may result a system into failure if not properly detected and recovered at time. These systems should perform with high convenience even under hardware and software package faults. Fault-tolerance is the important technique used to maintain reliableness in these systems. Hardware and software package redundancy are well-known effective strategies. Hardware fault-tolerance achieved through applying extra hardware like processors, communication links, resource (memory, I/O device) whereas in software package fault tolerance tasks, messages are added into the system to take care of faults. Fault should be detected by applying reliable fault detector followed by some recovery technique. Several fault detection techniques are offered however it's necessary to apply appropriate fault detector. Unreliable fault detector will build mistake by erroneously suspecting correct process or trusting crashed process.

II. TYPES OF FAULT

There are different types of fault which may occur in differing kinds Distributed System. These faults will be classified on many factors such as:

Network fault : A Fault occur in a network due to network partition, Packet Loss, Packet corruption, destination failure, link failure, etc.

Physical faults : This Fault occurs occur in hardware like fault in CPUs, Fault in memory, Fault in storage, etc.

Media faults : Fault happens due to media head crashes.

Processor faults : fault happens in processor because of software package crashes, etc.

Process faults : A fault that occurs because of shortage of resource, computer code bugs, etc.

Service expiration fault : The service time of a resource could expire while application is using it.

A fault is classified on the idea of computing resources and time. A failure occurs during computation on system resources is classified as: omission failure, timing failure, response failure, and crash failure.

Fault occurs with reference to time are :

Permanent : These failures occur by accidentally cutting a wire, power breakdowns and so on. It's easy to reproduce these failures. These failures will cause major disruptions and a few a part of the system may not be functioning as desired.

Intermittent : These are the failures seems appears. Mostly these failures are neglected whereas testing the system and only seem once the system goes into operation. Therefore, it's hard to predict the extent of damage these failures will bring back the system.

Transient : These failures are caused by some inherent fault within the system. However, these failures are corrected by retrying roll back the system to previous state like restarting package or resending a message. These failures are quite common in pc systems.

III. ISSUES

In a distributed real time system or in general, fault tolerance is the technique to provide the desired services in the presence of fault or error among the system. The aim is to avoid failures within the presence of faults and provide services as per demand. In fault tolerance the fault is detected initial and recovers them without participation of any external agents. The main issue in fault tolerance is how, where, and which technique is used to tolerate fault in distributed system. As we've seen several variety of fault and failure arises during a system, so there should be an acceptable technique which may tolerate such drawback. During this paper can we are going to see numerous technique for tolerating totally different fault.

In any real time distributed system there are 3 main problems.

1. *Feasibility* - This suggests that a task running should be finished on its deadline even if there's a fault within the system. Dead line in real time system is the major issue as a result of there's no meaning of such task that isn't finishing before its deadline. So the question is that which technique is to be applied by that the task will end on deadline within the presence of fault.

2. *Reliability* - In real time distributed system responsibleness means end of finish to end services and therefore the ability to experience failures or systematic attacks, without impacting customers or operations.

3. *Measurability* – It is regarding the flexibility to handle growing quantity of labour or, and therefore the capability of a system to increase total output below an increased load when resources are supplementary. Now the question arise however these faults will be detected and removed or tolerated from totally different setting. A task running in distributed setting should be finished on its deadline. It's going to be hard or soft rely on task requirement. In hard deadline a task should be finished by its deadline sharply however in soft deadline task will finished near its point in time.

IV. FAILED SYSTEM BEHAVIOR

What will happen if a system failed? What are the effects of a system failure? A system will behave after failure in 3 ways like

- Fail Stop System
- Byzantine System
- Fail-fast System

In fail stop system, there's no output once a system fails. It instantly stops to sending any message or event and also doesn't respond any message receiving on network. Any failure in a system in fail stop manner will results a permanent fault within the system.

Byzantine systems are those systems which not stop after failure but gives wrong output. These systems continue operating without desired output. They will send wrong output or may respond later results transient kind of fault.

In Fail-fast system the system behaves sort of a Byzantine system

for some time but moves into a fail-stop mode after a brief amount of time. It doesn't matter what kind of fault or failure has caused this behaviour but it's necessary that the system doesn't perform any operation once it has failed.

V. FAILURE DETECTION

Failure detection is the main issue in any system. Selecting a dependable failure detector is extremely troublesome task.

For detecting a fault accurately, a reliable fault detector is needed. It may be removed by applying appropriate removing techniques. Reliability of fault detector and fault recovery methodology is relying upon the sort of fault.

For removing a fault/failure from the system it's to be detected initial so fault tolerance technique is applied. Several failure detections are explained in numerous papers as Associate in Nursing freelance service . Several researchers have given fault detector for distributed system additionally. A failure detector should provide smart quality of service (QoS) however it is so far. Several schemes are applied however none of them resolve it properly. A failure detection service should adapt to dynamic network conditions and application necessities.

The failure detection service is enforced via variants of the Heartbeat mechanism. The heartbeat mechanisms are as follows: Centralized, Virtual ring based, All-to-all, and Heartbeat groups.

VI. APPROACHES OF FAULT TOLERANCE

There are many approaches for fault tolerance in real time distributed system. A fault can be tolerated on the basis of its behaviour or the way of

occurrence. Following are the methods of fault tolerance in a system.
1. Replication
a. Job Replication
b. Component Replication
c. Data Replication
2. Check-pointing

1. Replication

Replication, is the method of sharing data by which it will ensure consistency between redundant resources (i.e. package or hardware components), to enhance reliableness, fault-tolerance, or accessibility. Job replication is the methodology of replicating job on multiple server like in grid computing service is capable of receiving jobs, executing them, performing verification operations on them, and sending the result back to the consumer. A job can be distributed to servers for operation and result is back to the consumer. Data Replication is also usually utilized by fault tolerance mechanisms to reinforce accessibility in Grid like environments wherever failures are more likely to occur.
In this data is stored on multiple storage devices as a replica. Data replication may be synchronous or asynchronous depend on data consistency. Elements are replicated on completely different machines, and if any component or machine fail, then that application is transferred and run on another machine having the required elements.

2. Check – Pointing

It is the process to saving from complete execution of a task. It checks the acceptance test, if fail then go to previous checkpoint rather than starting. A check point is also system level, application level, or mixed level depends on its characteristics. Check-pointing is additionally categorised on the idea of In-transit or orphan message. These are Uncoordinated Check-pointing, Coordinated Check-pointing, and Communication-induced Check-pointing. Check-pointing can also be classified is based on World Health Organization instruments the application that do the particular capturing and re-establishing of the application execution state. These are Manual code insertion, Pre-compiler check pointing, Post-compiler check-pointing.

VII. CONCLUSION

To conclude this paper, we will say that it's very different to observe a fault in distributed system as compare with uniprocessor. Fault tolerance techniques are relying upon its occurrence. During this paper we have a tendency to explore various reliable fault detection and fault tolerance strategies. There are 3 things in real time distributed system that should be kept in mind once fault tolerance is applying. These are reliable, scalable and feasible.
In real time distributed system practicability of a task is far necessary as a result of there's a dead line outlined for every task and also the task should be finished on or before its deadline even there's a fault within the system.

REFERENCES

[1] en.wikipedia.org/wiki/**Fault_tolerance**
[2] web.info.uvt.ro/~petcu/distrib/
[3] www.cs.utexas.edu/lasr/download.php?uid=11
[4] en.wikipedia.org/wiki/**Fault_tolerance**