

# Review of Fault Tolerance Techniques in Distributed System

## First Author

Ayyaz Ahmed (16-MS-CE-09)  
Msc Computer Engineering  
UET Lahore,PAK  
ayyaz93@live.com

## Second Author

Baseer Hussain (16-MS-CE-1),  
Msc Computer Engineering  
UET Lahore,PAK  
baseer23@gmail.com

## Abstract

Distributed system is a collection of independent systems which can communicate with each other by transferring messages. There are some major issues in distributed systems but we focus in this paper on fault tolerance. It is the system's ability to work in the condition when there occur any type of some fault in the system, like failure in communication, hardware or resources. It is a very important issue in distributed system, in this paper we present a survey of different types of fault tolerance techniques and their comparison.

## 1 Introduction

A Distributed system is the collection of computers which are independent and appears to its user as a single coherent system. Distributed system is linked by some local networks and they are interlinked with each other's physically. It uses a computer network in which each computer works on some portion of overall task, in this manner a large task can be performed very efficiently and quickly than a single computer. Distributed System is much better than centralized system as it does not have any centralized controller so there is no chance of the failure in the distributed system because in distributed system computers are connected with number of servers so if one server is down it can get data from another server. It can easily extendable by adding more computers to the network. It allows many users to share data at the same time and makes man to man communication very easier.[6] Some Examples of distributed computing systems are online Air, Railway reservation systems, air traffic control systems, Cellular network, industrial control system, banking

system, wireless sensor network, multiplayer online games etc.

## 2 Issues in Distributed Computing System

Distributed system may have some issues and due to which its rate can be lower down. These issues can be:

### 2.1 Flexibility

The distributed system should be flexible so that the users can easily modify and enhance it.

### 2.2 Scalability

To design a system in such manner so that it can easily coping up and as a result growth of the system will increase. Central algorithms and central entities should be avoided by the system and most of the operations should be performed at the client work station by the system.

### 2.3 Fault Tolerance

The faults must be resisted by the system. If any fault may occur in future, the system performance doesn't degrade. The factors that generate faults in the system can be mobility, load imbalance, overloading and many more.

### 2.4 Security

The various resources of the system must be secured from unauthorized access and destruction so that the users can trust them. Due to the usage of insecure networks for data communications and the lack of its control from the single point, the provision of security in the distributed system is more difficult than in the centralized system.

As Now a days the size of distributed systems are increasing very rapidly day by day as the number of users are increasing, so the chance of problems and errors in the distributed systems

is also increasing. Millions of devices are working together and are prone to failures. Some examples of failures are memory, power, disks, processors and link failure. Diagnosing the faults in large distributed system is very difficult. It may halt or stop the normal working of distributed system and may turn the whole system execution in wrong path. In different reservation systems, internet banking, or in air traffic control system a single fault can give a very large amount of loss

### 3 Fault tolerance

Fault tolerance is the ability of a system to continue its working even in the presence of occurring any type of some faults. Fault tolerance can be get by the detection of that process in a system which are faulty, and then saving and restoring all of its computational works and then after this distributing the task which are recovered to all of the remaining process so that the system can continue to works. [1] Fault detector can prevent the loss due to any kind of the link failure or any kind of the failure in resource. Fault tolerance in the hardware can be get by adding some extra more hardware components which are faulty like memory, I/O device or processor.

#### 3.1 Need of fault tolerance

Fault tolerance techniques are needed due to the lot of reasons, some are

- Reliable processing of transaction
- Efficient working in case any fault occur
- Preventing any unauthorized access

To overcome the problem some fault tolerance techniques are needed which will enables the whole system to continue its working even in the presence of faults. There are lot of chances that one fault can arise or number of faults arises in the system simultaneous so simple fault tolerance techniques will not be enough to handle the situation and it will lead the system in wrong path and will not solve the purpose. So multiple fault tolerance techniques are needed to handle multiple faults which arise simultaneously.

## 4 VARIOUS FAULT TOLERANCE TECHNIQUES

We will discuss different fault tolerance techniques in this paper.

### 4.1 Replication Based Fault Tolerance Techniques

Replication based fault tolerance technique is one of the famous fault tolerance techniques. The word replica originates from the repetition and it means multiple copies. Replication is the process of maintaining different copies of same object or data item. In replication technique, client request is forwarded to one of replica among a set of replicas. [2] This replication technique is used for that request of client which will not alter the service state. Replication adds redundancy in the system. In this way if some of nodes fails to do work it will not result in the failure of the whole system and thus the fault tolerance is achieved as shown in fig 1

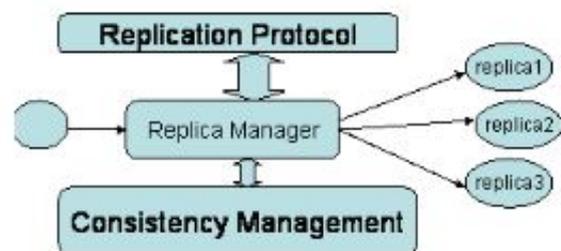


Figure 1: Replication Based Technique

Although Replication is a very good technique for fault tolerance but in this technique there are some issues that are Consistencies among all of the replica, replica on demand, replica management, degree of replica etc. We have discuss here consistency and degree of replica.

#### 4.1.1 Consistency

Consistency among replicas is the big issue in replication fault tolerance technique. Multiple copies of same data/entity causes the problem of consistency, the reason is update of any copy by any one of the users. A replication protocol must ensure the consistency between all of the replicas of the same object or entity. Consistency is ensured in this replication technique by some of the criterion. Many criteria for consistency have been defined such are sequential consistency and linearizability. In both the cases, an operation is performed on the most recent state of the object. [4] Sequential consistency informally states that a multiprocessor program executes correctly if its result could have been produced by executing that program on single processor

system. In order to have consistency an efficient strategy is required. Passive strategy and active strategy are main strategies. In a passive replication, only one primary execute requests and multicasts state changes to all replicas. This scheme avoids redundant computation of requests. It copes with non-deterministic service behavior. In active replica, client request is multicasts to all replicas. This means all replicas execute the request individually. In this way active replica takes less network resources than sending update. Active replica response to a fault is faster than passive. However, replica consistency usually requires deterministic replica behavior. [7]

#### 4.1.2 Degree of Replica

Number of replica is known as a degree of replication. In order to replicate an object a replication protocol is used. Primary-backup replication, voting, and primary-per partition protocol are some of the replication protocol. A replication protocol must be practical and simple. Niobe is such protocol purposed by researcher. Number of replicas must be sufficient. Large numbers of replicas will increase the cost of maintaining the consistency. Less number of replicas will affect the performance, scalability and multiple fault tolerance capability. Therefore, reasonable number replicas must be estimate as per system configuration and load. Researcher proposed adaptive replicas creation algorithm. There is further research scope to develop improved algorithm to maintain a rational replica number. Replica on demand is a feature that can be implemented to make more adaptive, flexible and dynamic. [3]

#### 4.2 Check pointing and Roll Back

Checkpoint with rollback-recovery is a well-known technique. Checkpoint is an operation which stores the current state of computation in stable storage. Checkpoints are established during the normal execution of a program periodically. This information is saved on a stable storage so that it can be used in case of node failures. The information includes the process state, its environment, the value of registers, etc. When an error is detected, the process is roll backed to the last saved state. Fig 3 shown below gives an idea about this technique

Basically this technique is used to restore

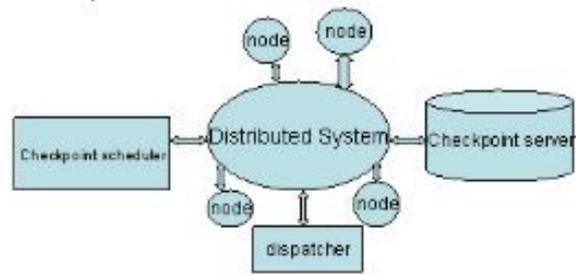


Figure 2: Check Pointing Technique

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. [2]

#### 4.3 Fusion Based Technique

Although replication method is widely used as a fault tolerance technique but number of backups is a main drawback. Number of backups increases drastically as coverage against number of faults increases. As the number of backup increases management of these backups is very costly. Fusion based techniques overcome this problem. It is emerging as a popular technique to handle multiple faults. Basically it is an alternate idea for fault tolerance that requires fewer backup machines than replication based approaches. In fusion based fault tolerance a technique, back up machines is used which is cross product of original computing machines. These backup machines are called as fusions corresponding to the given set of machines. Overhead in fusion based techniques is very high during recovery from faults. Hence this technique is acceptable if probability of fault is low. [6] Fig 4 below gives a basic idea about fusion

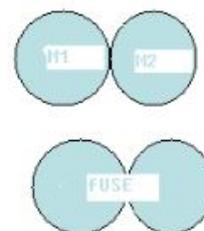


Figure 3: Fusion Based Technique

## 5 Comparison

Major Factors	Replication Based	Check point	Fusion Based
Working	Request is redirected to replica	State saved on Stable storage Used for recovery	Back up machine
Consistency	Linearizability, Sequential	avoiding updated messages	Among Back up machine
Multiple Faults Handling	Depends upon No of Replica	Depends upon Checkpoints Scheduling	Depends Upon No of machines

From the comparison it is clear that all methods having capability to handle multiple faults. In all methods performance can be improved by addressing the critical factors involved. In all techniques there is strong need for reliable, accurate and pure adaptive multiple failure detector mechanism.

## 6 CONCLUSION

Fault tolerance consists of two major components; failure detection and recovery. In this we have identified important issues such as fast, adaptive, accuracy, completeness, confidence and able to detect multiple faults. A reliable detector must not suspect a working process or processor and at the same time, it must detect all faults as early as possible. Multiple faults with performance is future trends of fault tolerance techniques Performance of a multiple fault tolerance algorithm depends on how much algorithm is capable to prevent the further loss due to faults.

## 7 References

### References

- Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation and Compiling*, volume 1. Prentice-Hall, Englewood Cliffs, NJ.
- American Psychological Association. 1983. *Publications Manual*. American Psychological Association, Washington, DC.
- Association for Computing Machinery. 1983. *Computing Reviews*, 24(11):503–512.
- Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the*

*Association for Computing Machinery*, 28(1):114–133.

Dan Gusfield. 1997. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, Cambridge, UK.