

# FUSION METHOD USED TO TOLERATE THE FAULTS OCCURRED IN DISTRIBUTED SYSTEM

Archana Thange<sup>1</sup>, Amina.N<sup>2</sup>, Amruta Nimbalkar<sup>3</sup>

<sup>1</sup>ME (IT), DKGIO'S COE, swami chincholi, Maharashtra, India

<sup>2</sup>ME (COMPUTER), DKGIO'S COE, swami chincholi, Maharashtra, India

<sup>3</sup>ME (COMPUTER), DKGIO'S COE, swami chincholi, Maharashtra, India

## Abstract

In a large distributed system there are more chances to occur faults, the only solution to prevalent this is to use the technique of replication. By using this technique some additional back-ups are needed. For example, if we have 'N' distinct data structures 'K' no of crash faults occurs each data structure by the use of back-ups, the replication technique needed 'K+1' replicates of each data structures. Here we present a solution of fusion method that combines the corrective codes and some selective replications to tolerate 'K' crash faults using 'K' additional fused back -ups. We show that our solution will give us  $O(n)$  savings in space instead of using replication. The theoretical and practical research results that the fused back ups are more space efficient as compared to replication because they causes minimal overhead for normal operation. Our solution requires only 120 back up structures instead of 300 back up structures which are used in replication method. This also results the power saving. In this paper we present a concept of using the combination of replication and fusion based technique.

**Keywords:** - Replication, fusion, primaries, back-ups, fault tolerance

-----\*\*\*-----

## 1. INTRODUCTION

Fault tolerance is an ability of a system to perform its functions correctly even in the presence of internal faults. The aim by using fault tolerance is to increase the dependability of a system. The approach to increase the dependability is fault prevention. This achieves by inspecting the system to eliminate the circumstances by which the fault arises [1].

**Error-** The reason or cause of failure is known as error. It represents an invalid state that is not specified in system behavior.

**Fault-** The real or root cause of a failure is known as a fault.

**Failure-** It is the symptom of a fault. Multiple failures are occurred due to a single error. Faults can be classified into two.

- 1) Transient
- 2) Permanent

Transient fault will be disappeared without any intervention. But a permanent fault will remain in the system if it is removed from an external agent.

Faults can be mainly classified into two types' crash fault and byzantine fault [6].

Crash fault is a type of fault in which the system either stops its complete work or it never returns to a valid state [5].

Byzantine fault is a type of fault of an arbitrary nature.

## 1.1 Basic Theory

We know that a distributed system have a structure of independent servers which are interacting with the clients by the use of messages. The data structures like linked lists, queues, hash tables are used to maintain large instance of data structures for the servers to efficiently store and manipulate the data [4]. These servers may become faulty by the data structures may crash, which leaves the total loss in its state leads to an arbitrary state, also which send some wrong messages to clients or other data structures. Normally we are using a technique known as replication. Suppose N given data structures are there and K no of crash faults occurs, the replication maintains K+1 replicas of each data structure resulting total of NK backups. For some large values of N, this is too costly in terms of space as well as power to maintain this back up process. Coding theory is the space efficient alternative to used as in replication process.

The data are transmitted across a channel in an encoded form by using some redundant bits that will correct the errors by the channels contain noisy. In a given set of data structures we maintain a set of fused back- ups that will tolerate 'K' crash faults among the given data structures. In replication process, the data structures replicated are identical to the data structures but in fusion there I no identical copy of given data structure are used. E can distinguish it as primaries and backups.

So, in our proposed system the data structures which are needed to tolerate the faults are known as primaries. And the back-ups contains only 'K' additional data structures. In replication it requires K+1 additional copies of the primary. The primary data's are in a coded form to save the space in

fused backups and to enable efficient updates by using replicate indexed structure of each primary.

**2. OBJECTIVE**

The main objective to introduce the fused backups to tolerate the faults occurred in distributed system. by using these fused backups which developer a system that reduces the fault recovery time. The maximum load can be distributed on any backups introducing this proposed system. Another motive to develop this proposed system is with space efficient and to reduce the updating time.

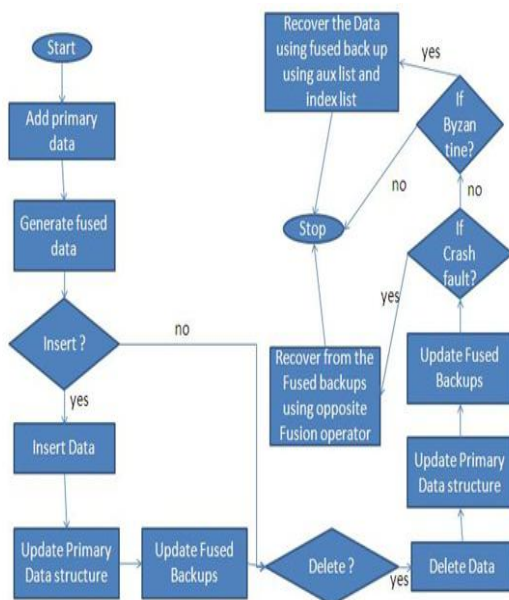
**2.1 Scope**

The main aim of the system developing is to reduce the overhead of replication. To tolerate 'K' crash faults among 'n' distinct data structures the technique of replication requires 'K+1' replicas of each data structure, which results 'NK' additional backups[8]. While using a fused data structure approach we use a fusion technique that uses a combination of erasure codes and some selective replication to tolerate 'K' crash faults using 'K' additional fused backups [3].

The fusion solution referred to use a combination of erasure codes and selective replicas to tolerate 'K' crash faults using 'K' additional fused backups [2]. The fusion solution saves O (n) space over replication.

**3. IMPLEMENTATION**

**3.1 Architecture of Proposed System**



**Fig 1:** architecture of system

The modules in the system contains

1. Formation of system
2. Primary data addition
3. Generation of fused data

4. Insertion of data
5. Deletion of data

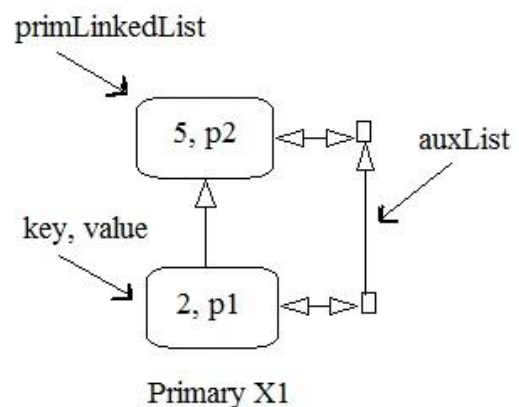
**Descriptions**

**3.1.1 Formation of system**

The 'N' given data structures are referred to a primaries X1... Xn.

The backup data structures are generated are referred as fused backups or fused data structures.

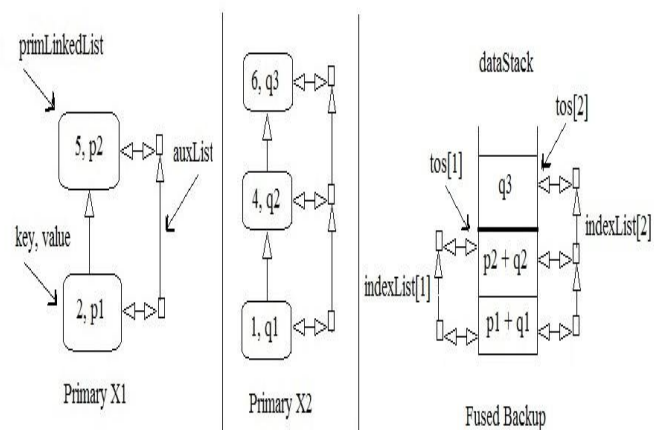
**3.1.2 Primary Data Addition**



**Fig 2:** Primary data addition

It is the module which is used to add the primary data to the server. It contains some key and value pairs. The figure shows the format of primary data addition.

**3.1.3 Generation of Fused Data**



**Fig 3:** Fused Data Generation

The fused data is generated with respect to the primary data. The backup data structures are generated based on the process of fusing primary data known as fused backups or fused data structures. The operator who combines the primary data is called fusion operator.

### 3.1.4 Insertion of Data

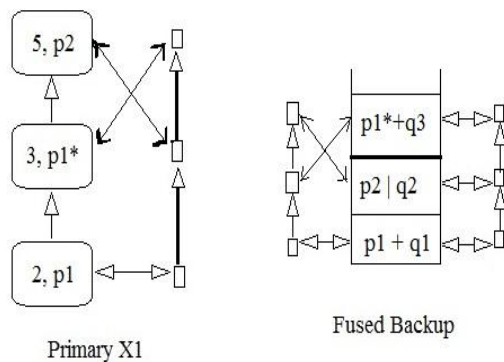


Fig 4: Insertion of Data

In this module if a client insert or update a data in the primary as well as fused data when he sends an insert to a primary X1, if the key is not already present, the primary X1 which creates a new node containing this value, insert it into the primary linked list and inserts a pointer to this node at the end of auxiliary list. The primary send key and the new key value is added and the old value which is associated with the key to all that fused backups. A data stack is used to maintain each fused data backup that contains primary elements in coded form. If the key value is not present on receiving the insert from X1 the backup data structure update the code value of the fused node following the one contains the topmost element of X1.

### 3.1.5 Deletion of Data

We can delete a key at the primaries and from the backups. The key value which contains the X1 from the primary which obtain its value which we want to send to the backups

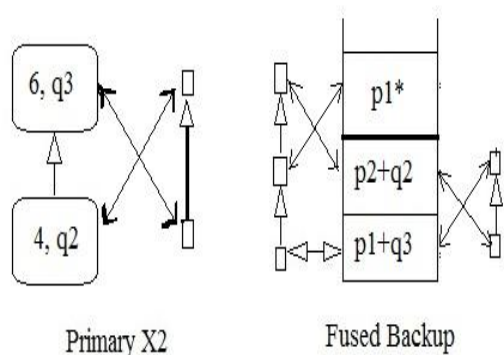


Fig 5: Deletion of Data

The primary can also send the value of the element which are pointed by the tail node of the auxiliary list along with the value and the key 'K'. The topmost element of X2 is shifted to the backup stack. After the sending operation the final node value of the auxiliary list is shifted by the primary to the position of the auxiliary node pointing to the deleted element.

## 4. CONCLUSIONS

The method of fusion technique for fault tolerance that saves as  $O(n)$  savings in space while comparing to the replication method without any overhead during its normal operation. The proposed system will provide us the space optimality, efficient updates and updating of optimality and order of independence. We can use this system to reduce the data structures required to recover the faults, to increase the efficiency of fault tolerance in data structures. It can be used in all the data structures as a cost effective fault tolerance method.

## REFERENCES

- [1]. Bharath Balasubramanian and Vijay K. Garg, Fellow, IEEE, "Fault Tolerance in Distributed Systems Using Fused Data Structures", IEEE transactions on parallel and distributed systems, vol. 24, no. 4, April 2013
- [2]. B. Balasubramanian and V.K. Garg, "Fused Data Structure Library (Implemented in Java 1.6)," Parallel and Distributed Systems Laboratory, <http://maple.ece.utexas.edu>, 2010.
- [3]. J. Blomer, M. Kalfane, M. Karpinski, R. Karp, M. Luby, and D. Zuckerman, "An XOR-Based Erasure-Resilient Coding Scheme," Technical Report TR-95-048, Int'l Computer Science Inst., Aug. 1995.
- [4]. V.K. Garg and V. Ogale, "Fusible Data Structures for Fault Tolerance," Proc. 27th Int'l Conf. Distributed Computing Systems (ICDCS '07), June 2007.
- [5]. V. Ogale, B. Balasubramanian, and V.K. Garg, "A Fusion-Based Approach for Tolerating Faults in Finite State Machines," Proc. IEEE Int'l Symp. Parallel and Distributed Processing (IPDPS '09), pp. 1-11, 2009.
- [6]. V.K. Garg, "Implementing Fault-Tolerant Services Using State Machines: Beyond Replication," Proc. 24th Int'l Conf. Distributed Computing (DISC), pp. 450-464, 2010.

## BIOGRAPHIES



Archana Thange is a student of M. E. (I.T) second year. She has completed her B.E. in Information Technology. She has worked as an assistant Professor at H.S.B.P.V.T's COE kashti.



M. E. (Computer) second year. She has completed her B.Tech. In Computer Science. She has worked as an assistant Professor at H.S.B.P.V.T's COE kashti.

M. E. (Computer) second year. She has completed her BE in Information Technology.