A SINGLE DISK FAILURE RECOVERY FOR X-CODE BASED PARALLEL STORAGE SYSTEMS

M. Nithya

Department of Computer Science and Engineering, Raja College of Engineering and Technology, Madurai, Tamilnadu (India)

ABSTRACT

Achieving data availability and reliability guarantees against disk failure using redundancy coding scheme. X-code is a double tolerant to achieve the optimal update complexity. It reduces the possibility of data unavailability when disk/node fails. X-code based optimal recovery scheme Minimum-disk-read-recovery (MDRR) and Group-based MDRR (GMDRR) minimizes the number of disk reads for a single-disk failure recovery. A tight lower bound of disk read is formed and MDRR algorithm is applied to match the theoretical lower bound. A disk read cannot be balanced while matching the lower bound of disk reads within a single stripe and cannot be balanced among different disk by simply rotating disk. A leap rotation scheme which balances disk read among different disk within a group of stripes and it matches the lower bound of disk read is called as GMDRR. MDRR reduces around 25% percent of recovery time of the conventional approach.

Keywords: Distributed System, Data Availability, Recovery Algorithm

I.INTRODUCTION

A large parallel storage system is to make information reliable and available for a long period of time. To achieve reliability and availability in the face of component failures redundancy techniques have been widely used in parallel storage. RAID-6 array codes can be categorized into two classes. The first class is the horizontal codes, such as RDP and EVENODD, where original data fragments are stored in data disks while encoded fragments (known as parities) are stored in dedicated parity disks (also known as P and Q disks). Another class is vertical codes, such as X-code, where parities are distributed across all disks. When a disk failure occurs in a parallel storage system, it is important to recover the erased data in the failed disk as quickly as possible to maintain the system reliability guarantees.

A conventional approach is to download the entire file and reconstruct the original data, and then regenerate the data fragments of the failed disk. The conventional approach will cost a great deal of data transmission. The total amount of data that must be processed during recovery plays a crucial role in recovery time and affects the system service performance. This is particularly important in parallel storage systems, where network transmission bandwidth is a potential performance bottleneck.

II. RELATED WORK

The constructions of new MDS codes can achieve improved recovery performance. Hu et al. further consider cooperative recovery for multi-node failures. However, regenerating codes have yet to see practical deployment, possibly due to several constraints. Most regenerating codes require storage nodes be programmable to support the encoding capability for recovery, thereby limiting the deployable platforms for practical storage systems. Some regenerating codes can be implemented without the encoding capability of a storage node, but generally introduce higher storage overhead than traditional erasure codes. Their encoding operations involve linear computations on finite fields, and are more computationally expensive than XOR-based MDS array codes. An efficient recovery scheme called the Path Directed Recovery Scheme (PDRS), which can decrease the disk I/O complexity by upto 25 percent for all vertical RAID-6 codes like P-code and X-code when recovering a single failed disk. However, they do not formally derive the lower bound of disk reads. Also, PDRS cannot consider the load balancing problem among different disks during recovery. Its performance decreases greatly as the size of storage system increases as their experiments indicated.

III. PROPOSED SYSTEM

The core objectives of the proposed system is to recover a single disk failure for parallel storage systems using a well-known MDS array code called X-code, which can tolerate double disk failures X-code is optimal in computational complexity, update complexity, and space efficiency among all the RAID-6 codes Unlike RDP and EVENODD codes, both of which are horizontal codes, X-code is a vertical code that has a different geometrical structure where parities are placed in rows rather than columns. Thus, X-code has an advantage of achieving load balancing for data updates within a stripe among different disks, instead of aggregating parities in dedicated parity disks as in RDP and EVENODD codes. Due to the different geometrical structure, the recovery algorithms previously proposed for RDP and EVENODD codes are no longer applicable for X-code.

The process of Fault Tolerance and Recovery process is done by MDRR and GMDRR Algorithm. MDRR does not possess the disk read balancing property, i.e., it reads different number of symbols from different disks. In case of unbalanced disk read, a disk with a heavier load will slow down the recovery and degrade the availability of the system. Here, we prove that disk read cannot be balanced in a stripe while matching the lower bound in general cases. Furthermore, it cannot be balanced by disk rotation. We then present a method which balances disk read in a group of p -1 stripes while matching the lower bound. MDRR (Minimum Disk Read Recovery) algorithm will extract the n-number of disk node and evaluate the availability of disk. The availability of disk is valid, the generate parity strip will be submitted otherwise search other disk node. In GMDRR creates group of parity rows of specific disk node. The generated group of rows is split by the EVENODD Row Algorithm. The separated group EVEN or

ODD rows will transfer valid disk arrays. The GMDRR and EVENODD Algorithm will be used recovery for the failed strip of valid disk.

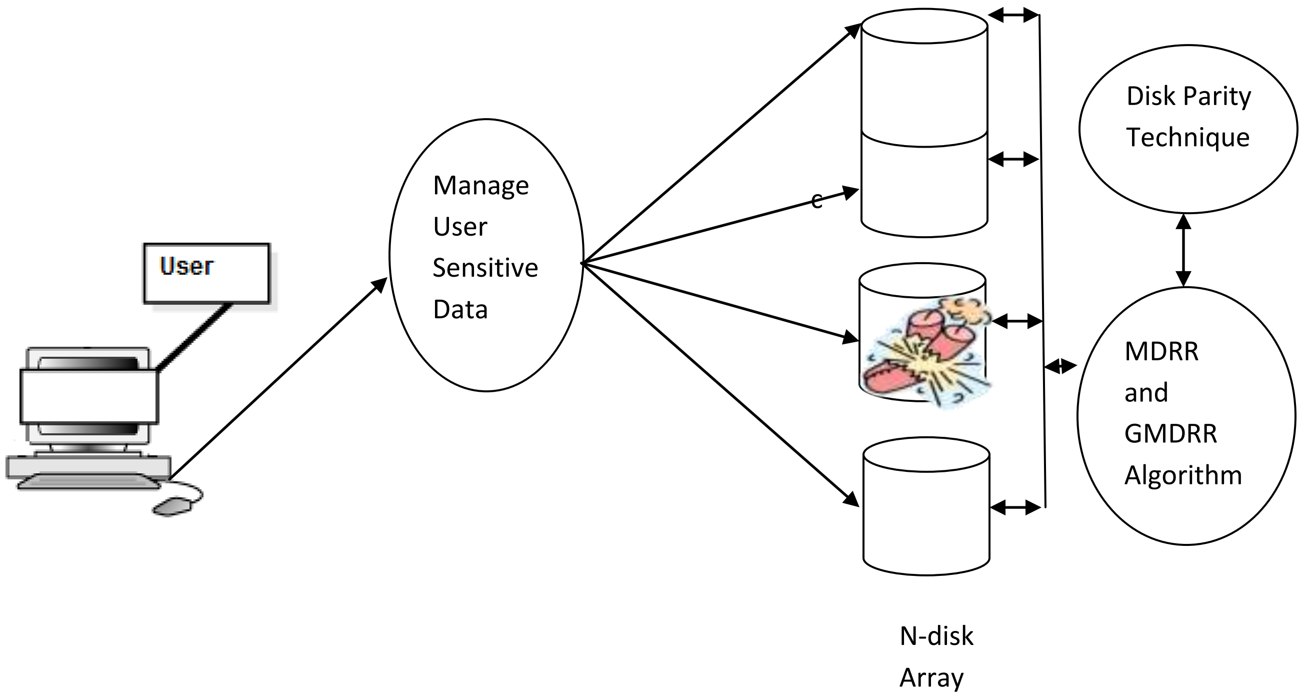


Figure 1: System Diagram

IV. RESULTS AND DISCUSSSIONS

The process of sensitive data management and storing it in n-number of disk by credit card based business application. Five nodes are taken and the business application is created. The business application having following process,

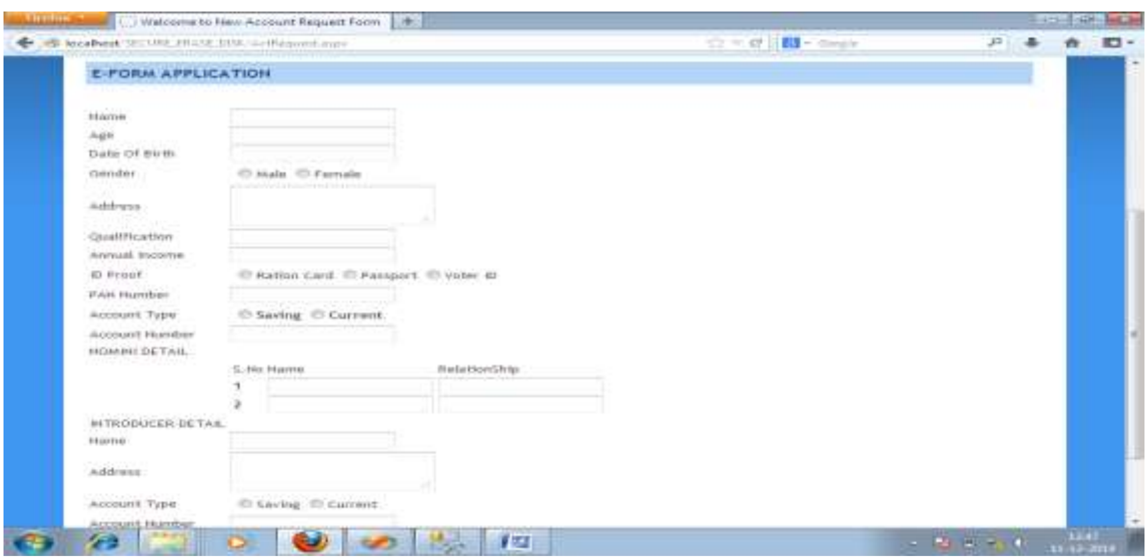
- a. Account Request
- b. Activation
- c. User Transaction
- d. Company Service
  - i. Disk Parity Service
  - ii. Recovery Console

Account Request form is managing the new customer account service and activation is used activate user requested account, providing unique ID for business transaction. The account activated customer start business transaction and submits sensitive data to server. Each and every transaction rows are transferred n-number of storage disk. If any fault occurs for data storage, RAID Technique is applied and transfers the user sensitive to available storage node. The company module is used to manage the disk parity service to count the rows and replace the other storage with the help MDRR and GMDRR Algorithm.

BUSINESS APPLICATION

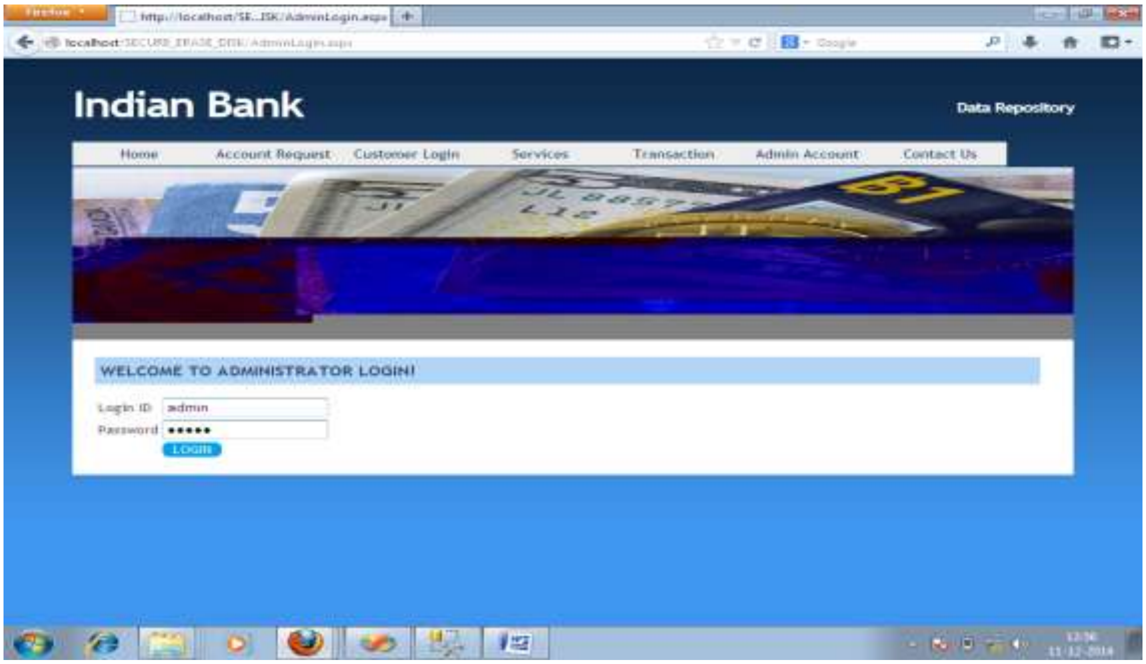
Bank application is created to explain the disk/node failure. This module consists of user transaction, administrator maintains the user account and their transaction. Row parity is maintained by the transaction log.

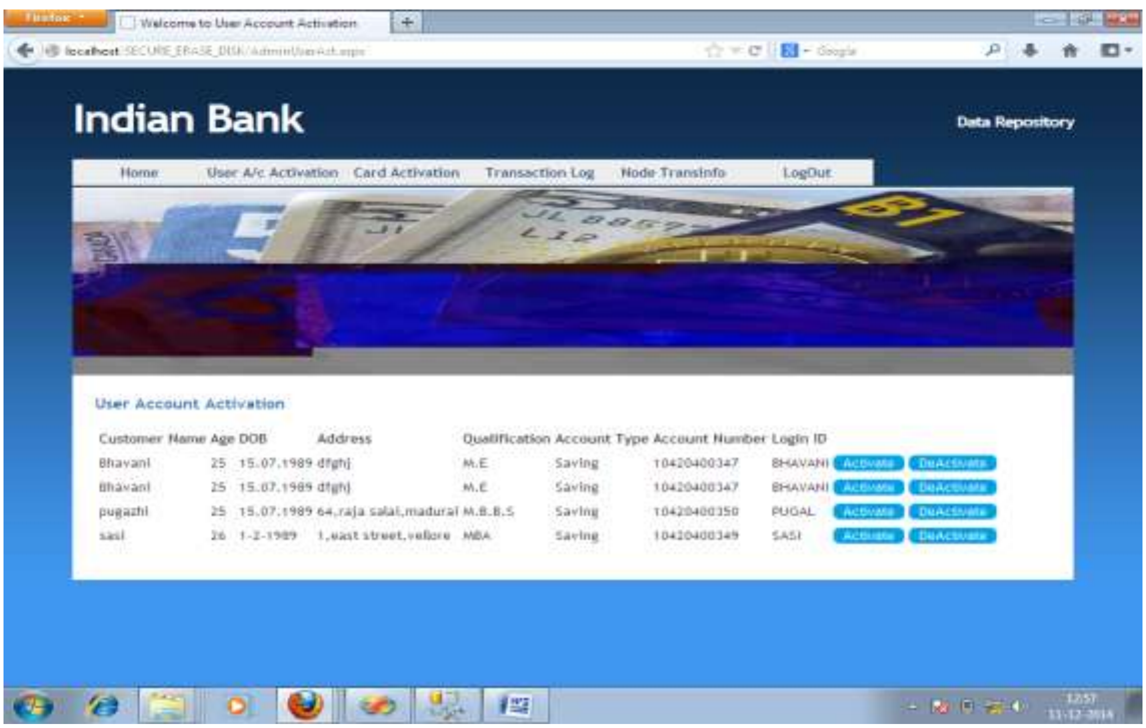
ACCOUNT REQUEST FORM



User opens an account in the bank by providing their personal information. Bank automatically generates the account number for user.

ADMINISTRATOR LOGIN FORM

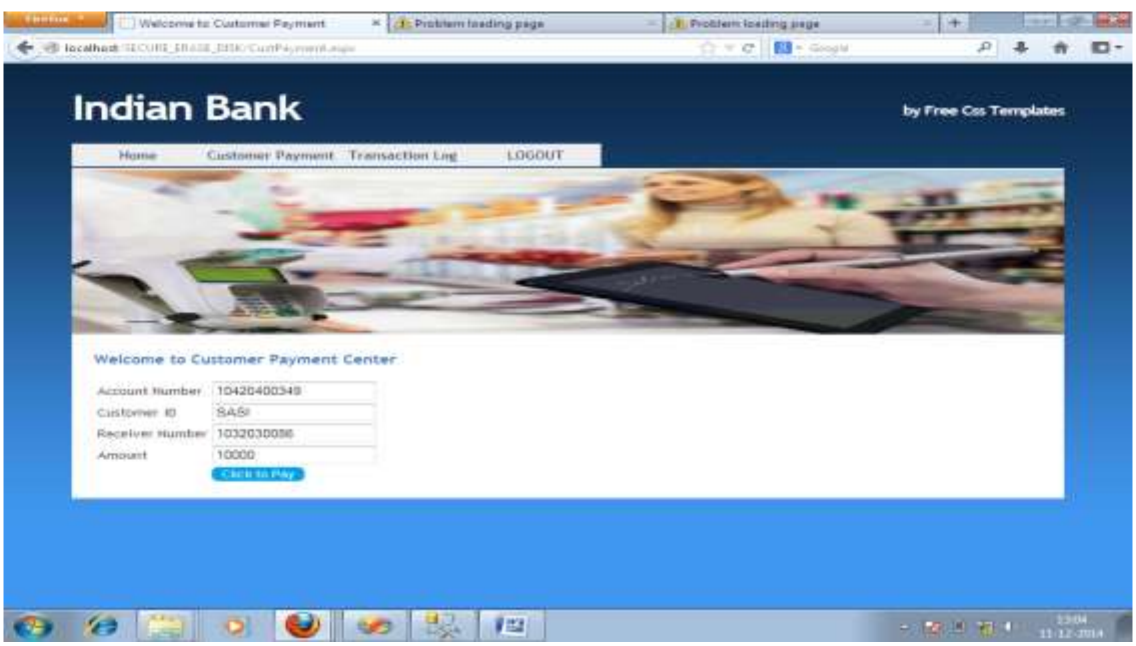


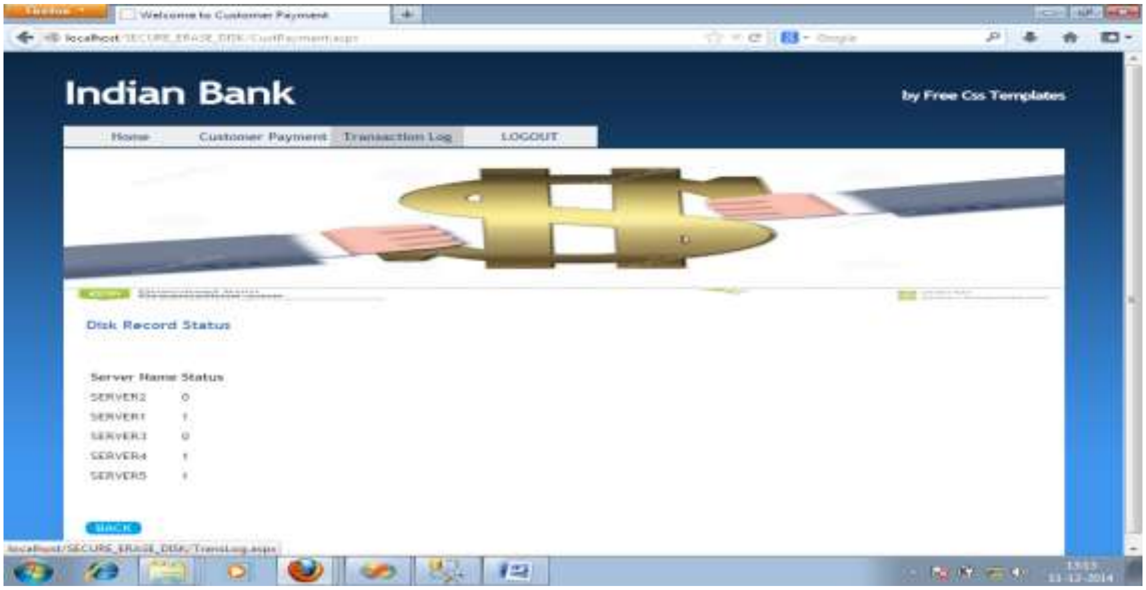


Administrator activate the user account, maintain the transaction made in each disk and provides the node transformation information.

CARD ACTIVATION

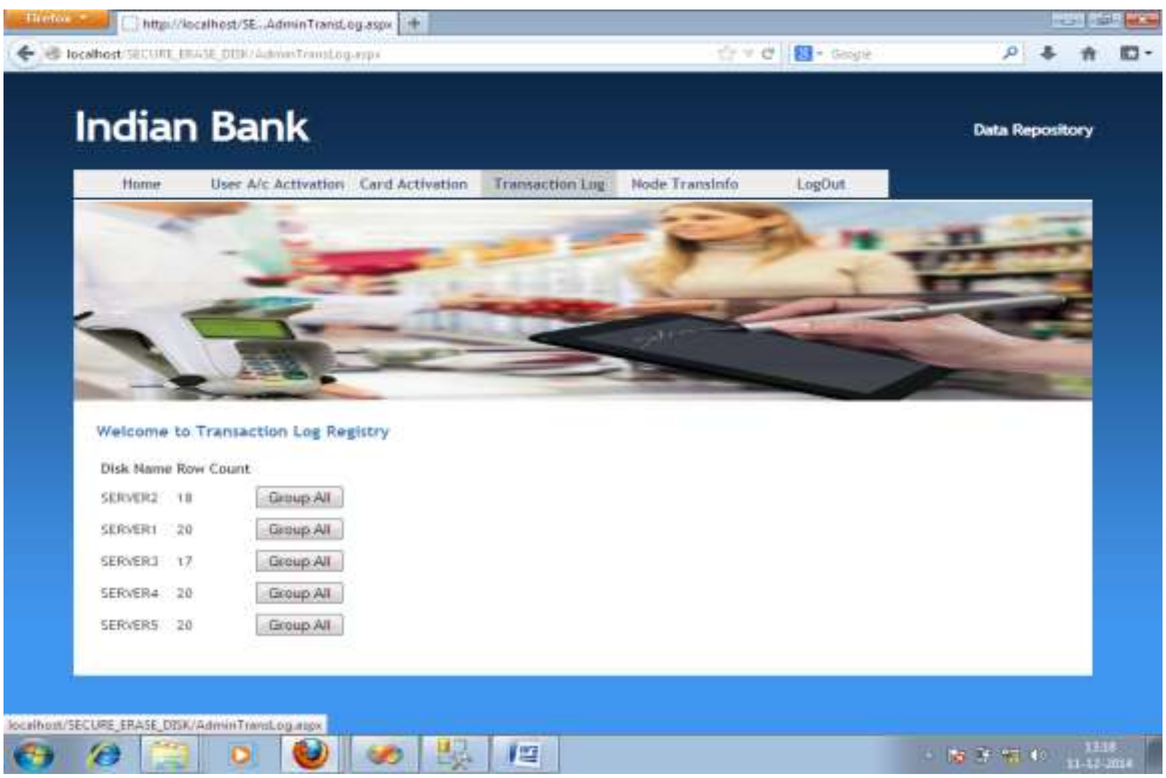
CUSTOMER TRANSACTION





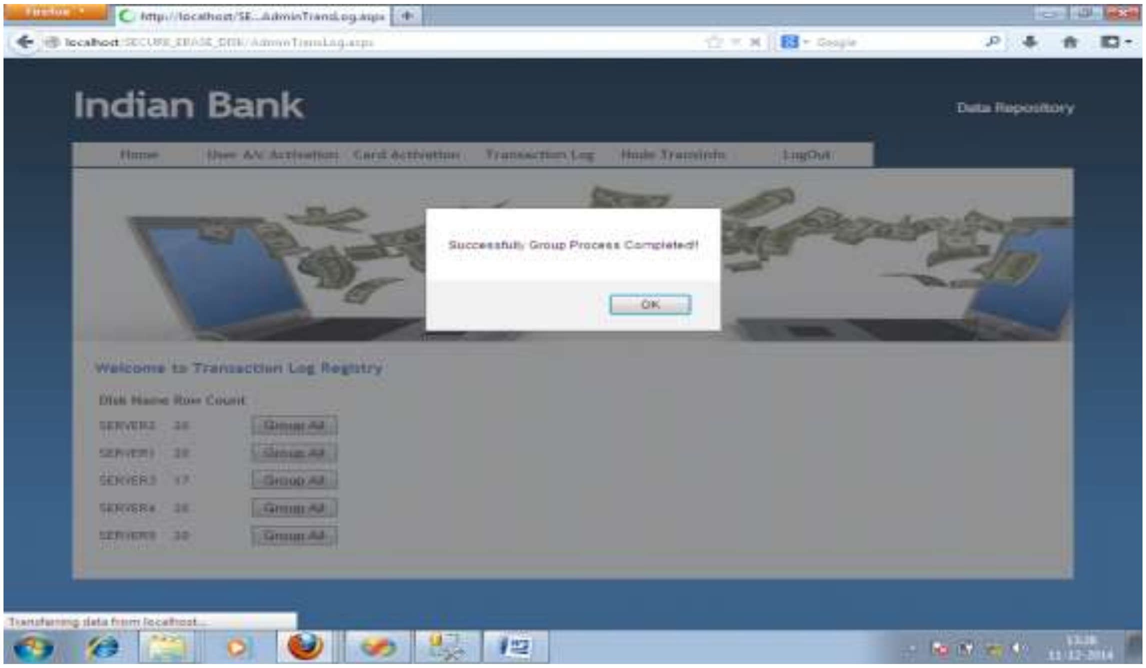
Transaction is made by the user by specifying the receiver account number and the amount. User transaction is viewed in each disk using this module. Disk2 and Disk3 is failed. So the status of the disk 2 and disk 3 is denoted as 0.

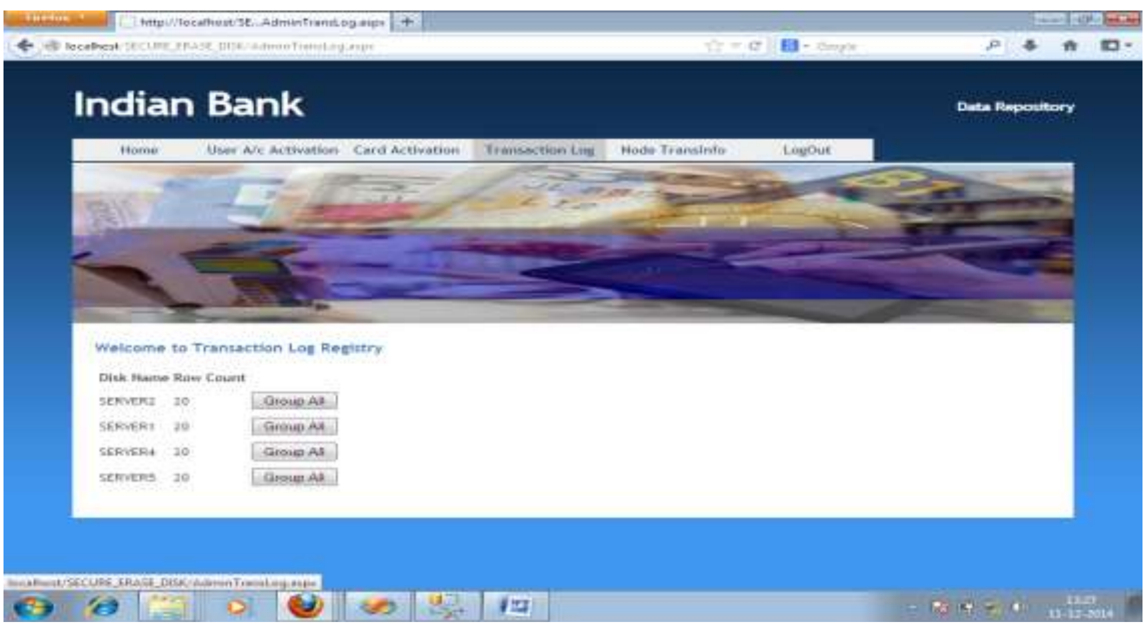
TRANSACTION DETAILS



The detail of each disk is specified in this module i.e no of transaction made in each disk. Disk2 has failed in last two transaction and disk3 has failed in last 3 transaction. We can retrieve the data of Disk2 and Disk3 when it is available in network.

ROW PARITY GROUPING





Disk2 have been retrieved by the last two data lost. We are grouping the disk which has a maximum element from the remaining disk.

V. CONCLUSION

Optimal recovery solutions are mainly designed for RAID-6 horizontal codes, this is the first work that addresses the optimal recovery problem of RAID-6 vertical codes. The principle of leap rotation is implementing data encoding based on logical number of disks, and this rotation scheme can be applied to balance disk reads in storage systems with different codes. The problem of existing techniques is finding the maximum of record count in all disk arrays and create group to identifying availability of record. The created group will be replaced with the help of MDRR and GMDRR algorithm. The process take long time to fill in rows of all disk. But our Dump techniques eliminate the above problem and reduce time interval for fill into rows at disk arrays.

REFERENCES

[1] N.B. Shah, K.V. Rashmi, P.V. Kumar, and K Ramchandran, “Regenerating Codes for Distributed Storage Networks,” Proc.Int’l Conf. Arithmetic of Finite Fields, pp. 215-223, 2010.

[2] Y. Hu, Y. Xu, X. Wang, C. Zhan, and P. Li, “Cooperative Recovery of Distributed Storage Systems from Multiple Losses with Network Coding,” IEEE J. Selected Areas in Comm., vol. 28, pp. 268- 276, Feb. 2010.

[3] C. Suh and K. Ramchandran, “Exact-Repair MDS Codes for Distributed Storage Using Interference Alignment,” Proc. IEEE Int’l Symp. Information Theory, 2010.

[4] K.V. Rashmi, N.B. Shah, P.V. Kumar, and K. Ramchandran, “Explicit Construction of Optimal Exact Regenerating Codes for Distributed Storage,” Proc. Allerton Conf., 2009.

[5] Z. Wang, A.G. Dimakis, and J. Bruck, “Rebuilding for Array Codes in Distributed Storage Systems,” Proc. Workshop Applicatio of Comm. Theory to Emerging Memory Technologies (ACTEMT), 2010.

[6] L. Xiang, Y. Xu, J.C. Lui, Q. Chang, Y. Pan, and R. Li, “A Hybrid Approach to Failed Disk Recovery Using RAID-6 Codes: Algorithms and Performance Evaluation,” ACM Trans. Storage, vol. 7 no. 3, pp. 11:1-11:34, Oct. 2011.

[7] Q. Cao, S. Wan, C. Wu, and S. Zhan, “An Evaluation of Two Typical Raid-6 Codes on Online Single Disk Failure Recovery,” Proc. IEEE Fifth Int’l Conf. Networking, Architecture and Storage (NAS), pp. 135-142, 2010.

[8] S. Li, Q. Cao, J. Huang, S. Wan, and C. Xie, “PDRS: A New Recovery Scheme Application for Vertical Raid-6 Code,” Proc. IEEE Sixth Int’l Conf. Networking, Architecture and Storage (NAS), pp. 112-121, 2011.

[9] O. Khan, R. Burns, J.S. Plank, W. Pierce, and C. Huang, “Rethinking Erasure Codes for Cloud File Systems: Minimizing I/O for Recovery and Degraded Reads,” Proc. 10th USENIX Conf. File and Storage Technologies, p. 20, 2012.

[10] Y. Zhu, P.P.C. Lee, Y. Hu, L. Xiang, and Y. Xu, “On the Speedup of Single-Disk Failure Recovery in XOR-Coded Storage Systems: Theory and Practice,” Proc. IEEE 28th Symp. Mass Storage Systems and Technologies, 2012.