IJCTA, 9(10), 2016, pp. 4257-4267

A Dual Quorum-Reed Solomon Coded Protocol Handling Hybrid Failures in Distributed Storage

***R.R. Sharanya ** C.K. Shyamala**

Abstract : It is highly important to design a reliable Distributed Storage System (DSS) in the age of increased growth of data. Secure and reliable storage of data have become the top priority for all organizations dealing with high volume of data. To provide availability, scalability and fault tolerance, Replication and/or Dispersal technique is adopted in DSS. The presence of hybrid failures in DSS highly affects its performance. To design a reliable DSS, hybrid failure should be modelled and handled. This paper explores how a Dual quorum (DQ) protocol can be integrated with Dispersal technique to provide security (confidentiality) in addition to the fundamental DSS features. The paper analyses the performance of DQ-dispersal with that of DQ-replication in the presence of hybrid failures.

Keywords : Availability, Fault tolerance, Dual Quorum, Hybrid failures, Reed Solomon Codes.

1. INTRODUCTION

In the digital world, most of the data are unstructured which are generated by all our digital interactions and the gadgets like smartphones, camera, sensors etc. leads to the explosion of data. Businesses can store these data in their own data centers or they can get service from the data storage providers. DSS are capable of handling these data quickly and efficiently. Fault tolerance is an important factor that every storage system should provide. In order to provide fault tolerance, either replication and/or dispersal technique is employed in DSS. Most of the present systems are employed with replication technique which does not provide inherent security at DSS; The best alternative is to incorporate dispersal at DSS. This paper proposes a solution that integrates DQ with dispersal for hybrid failures in the operating environment. The paper is organized as follows; Section 2 discusses about related works. Section 3 presents the proposed architecture and design of the system. Section 4 compare and analyses the performance of DSS for RS based dispersal. Section 5 concludes the paper.

2. RELATED WORKS

Quorum systems are employed to cope up with network partitioning. A quorum system is a collection of subset of nodes called quorum with a property that quorums have a non-empty intersection. An operation/transaction is performed if and only if the majority of node in the quorum reach a consensus. Quorum systems are used to ensure consistency in DSS in addition to integrity and availability. The client reads from read server, r and perform write operation to the write servers, w where $r + w > n$. The latest updated value is obtained from the non-empty intersection. In the traditional quorum system, when there is a simultaneous read and write request the availability

* Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India Sharanyarr93@gmail.com

** Department of Computer Science and Engineering, Amrita School of Engineering, Coimbatore, Amrita Vishwa Vidyapeetham, Amrita University, India ck_shyamala@cb.amrita.edu

is at risk. Quorum based systems provide remarkable fault tolerance despite availability issue [2] [17] [18]. To resolve this problem, Dual Quorum is introduced in [2] [12] There are two dedicated quorums say Q_{output} and Q_{input} systems to handle client's read and write requests respectively to overcome the risk of availability faced in basic quorum systems. A file of size, |F| is partitioned into*n* fragments (parts with redundancy) such that the original file, F can be reconstructed from a minimal success of *k* fragments. This process is called data dispersal. Many Dispersal algorithms such as Information Dispersal Algorithm (IDA) [6] [8] [9] [17] [21], Reed Solomon Codes (RS Codes) are employed in the DSS to improve the storage efficiency and provide better security compared to that of the DSS that employed Replication technique [3] [10] [11] [12] [13]. Using these dispersal algorithms, the data is sliced into many pieces and stored across multiple nodes to provide fault tolerance in the DSS. The problem in erasure coding such as IDA is their ability to tolerate only erasures whereas error correcting codes such as RS Code is able to tolerate both errors and erasures at DSS.The major concern while designing a reliable DSS is the failure. Failures can be categorized into 2 types namely, Crash and Non-Crash failure. In case of crash failures the system will halt but is working properly until it halts. This defines the Fail Stop case. Whereas non-crash failure, the system will deviate from the expected behavior. The combination these two failures are termed as Hybrid failures. The important reason for hybrid failure is the presence of adversary in DSS. An adversary may be a standaloneentity or group of entities that pose threat(s) to the security of a system or group of systems.There are different types of adversary and are highly discussed in [16].

3. PROPOSED METHOD

A. Dual Quorum Dispersal

Several issues like fault tolerance, availability, storage efficacy have to be taken into consideration while designing DSS [1] [15]. Naïve replication and Quorum based replication approaches [2] lack in security in particular data confidentiality. In replication based storage, if one copy is revealed to an illegitimate entity, confidentiality of the original data is lost in its entirety. This issue is overcome in [12]; RS coded data can be retrieved iff *k* of the data fragments are made available for reconstruction. An illegitimate entity has to compromise *k* servers for obtaining the original data. Dual quorum protocol with dispersal technique (DQ-D) is integrated in [12] which provides availability, efficient storage and improved security (confidentiality), under the assumption of *fail stop* condition. Fail stop assumes that once a node fails, no data will be sent from that node to other nodes further. The architecture illustrated in Fig.1. has been enhanced for dispersal from the design in [2].

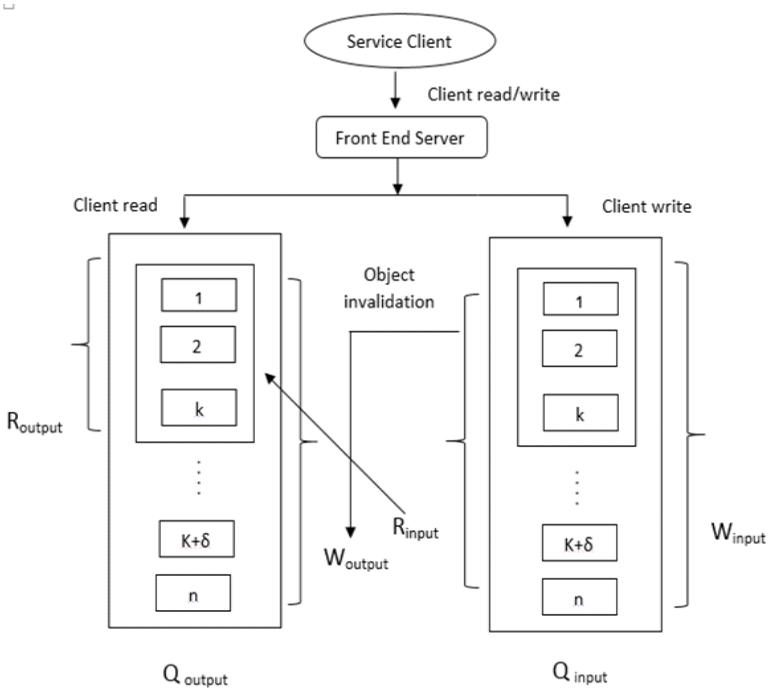

Fig.1. Dual Quorum-Dispersal Architecture.

B. Read and Write Operation.

To perform a successful *Read* operation in DQ-D system, data (fragment) has to be retrieved from *k* servers and for a successful *Write* operation, data has to be written into *n* servers as illustrated in Fig.2. Four possibilities arise - Read Hit, Read Miss, Write Through and Write Suppress. When a Read request arrives, Front End Server (FES) redirect the request to Q_{output}. If the requested fragment is available then it is called ***Read Hit***. The read hit is showed in Fig.2.

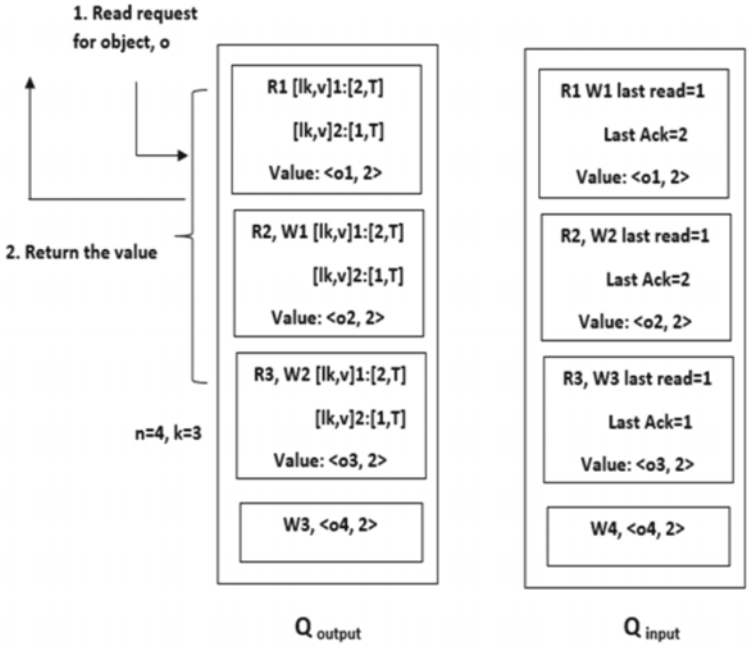

Fig. 2. Read Hit.

A Read Missoccurs when the fragments in the Q_{output} are not valid. In this case the fragments with highest timestamp will be retrieved from the Q_{input} as illustrated in Fig.3.

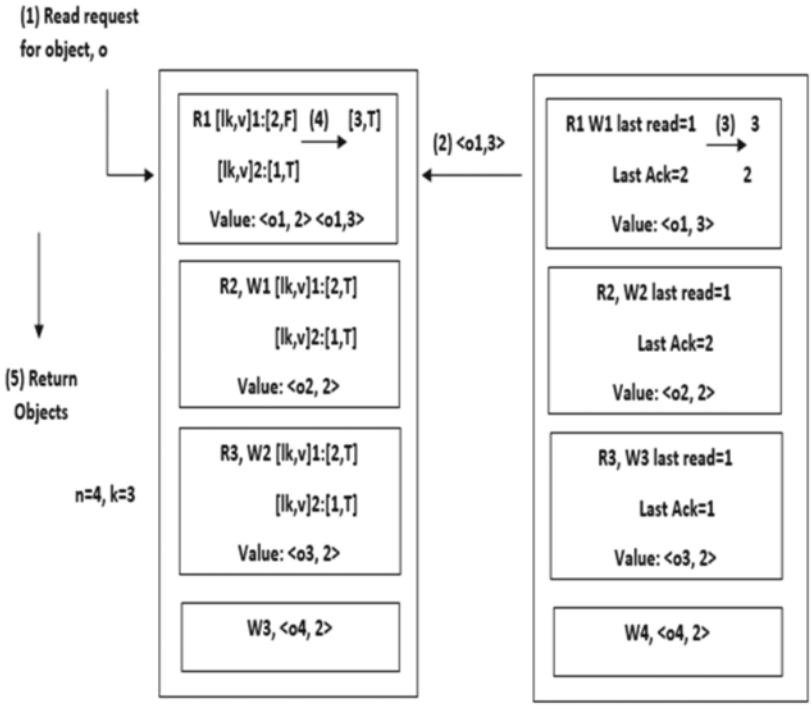

Fig. 3. Read Miss.

1. Pseudocode for Read

```
For every read request
{
  Step1: Forward the request to read servers of Q output system
  Step2: Randomly select 'k' servers out of 'n' to obtain data (fragments
  Step3: Check the validity of fragments
      {
          If (valid = true)
          {
           Case = Read Hit;
           Perform Read;
          }
           Else if (valid = false)
           {
            Case = Read Miss;
            Get latest fragment from R input server;
            Update last known in Q output system;
            Make valid to True;
            Perform Read.
           }
      }
}
```

In write through, the fragments have to be invalidated at the Q_{output} before performing the write operation as depicted in Fig.4.

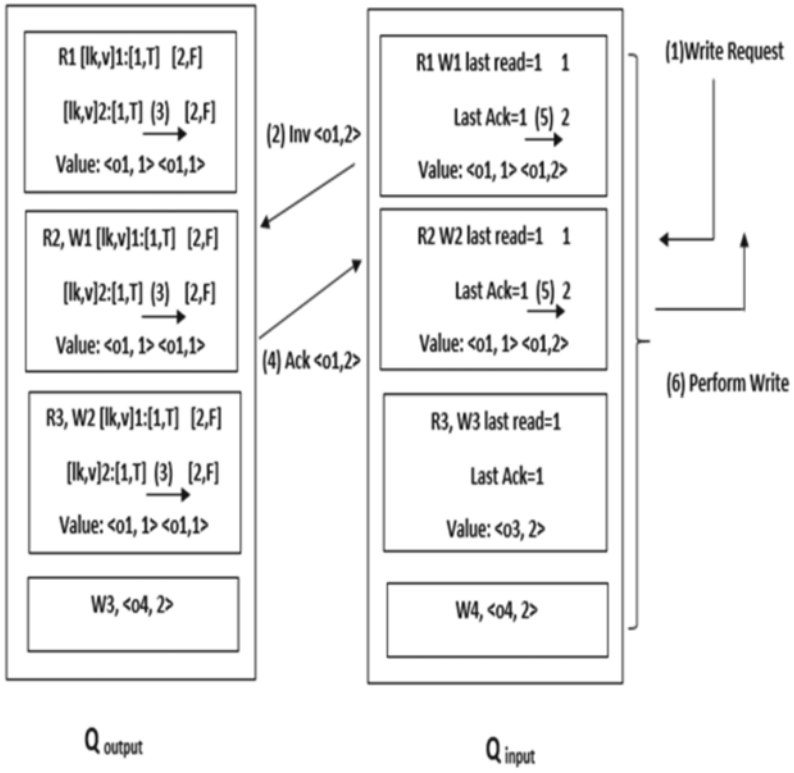

Fig. 4. Write Through.

In write suppress, the write is performed without communicating with Q_{output} system as the invalidation is done already. [2] as illustrated in Fig.5

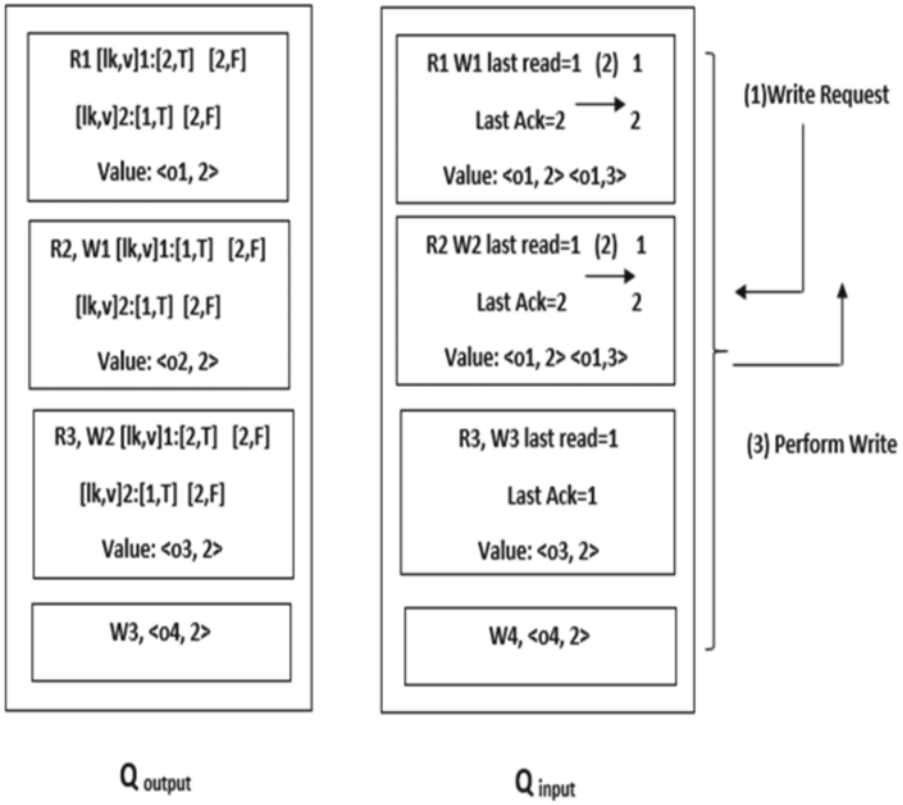

Fig. 5. Write Suppress.

2. Pseudocode for Write

```
For every write request
{
Step1: Forward the request to write servers of Q input system.
Step2: Check
        If (last read < last ack)
            {
            Case = Write Suppress;
            Perform Write.
            }
        Else if (last read > last ack)
           {
            Case = Write Through;
            Make valid = false in read servers of Q output system;
            Get acknowledgement from Q output system;
            Perform Write.
           }
}
```

4. DQ-D WITH RS CODED STORAGE

RS Codes is a special form of Error Correcting Codes. The encoder takes *k* data symbol and adds*m* parity symbols to make an *n* symbol code word. It is represented by RS(*n, k*) where *n* is the total number of fragments and *k* is the minimum number of fragments required to reconstruct the data. The decoder can correct up to *t* symbols that are errors and $2t = n - k$ erasure. [7] [14] RS code is robust as it handles both errors and erasures as depicted in Fig.6 which is very much important in the design of DSS.Read operation can be performed if any of the *k* servers out of *n* servers in Q_{output} system are correct. Data is reconstructed from *k* servers which may be available and correct or available but corrupted. If any of the *k* server is under an adversary control, those servers are corrected and the fragments are restored from incorrect and/or invalidated servers.

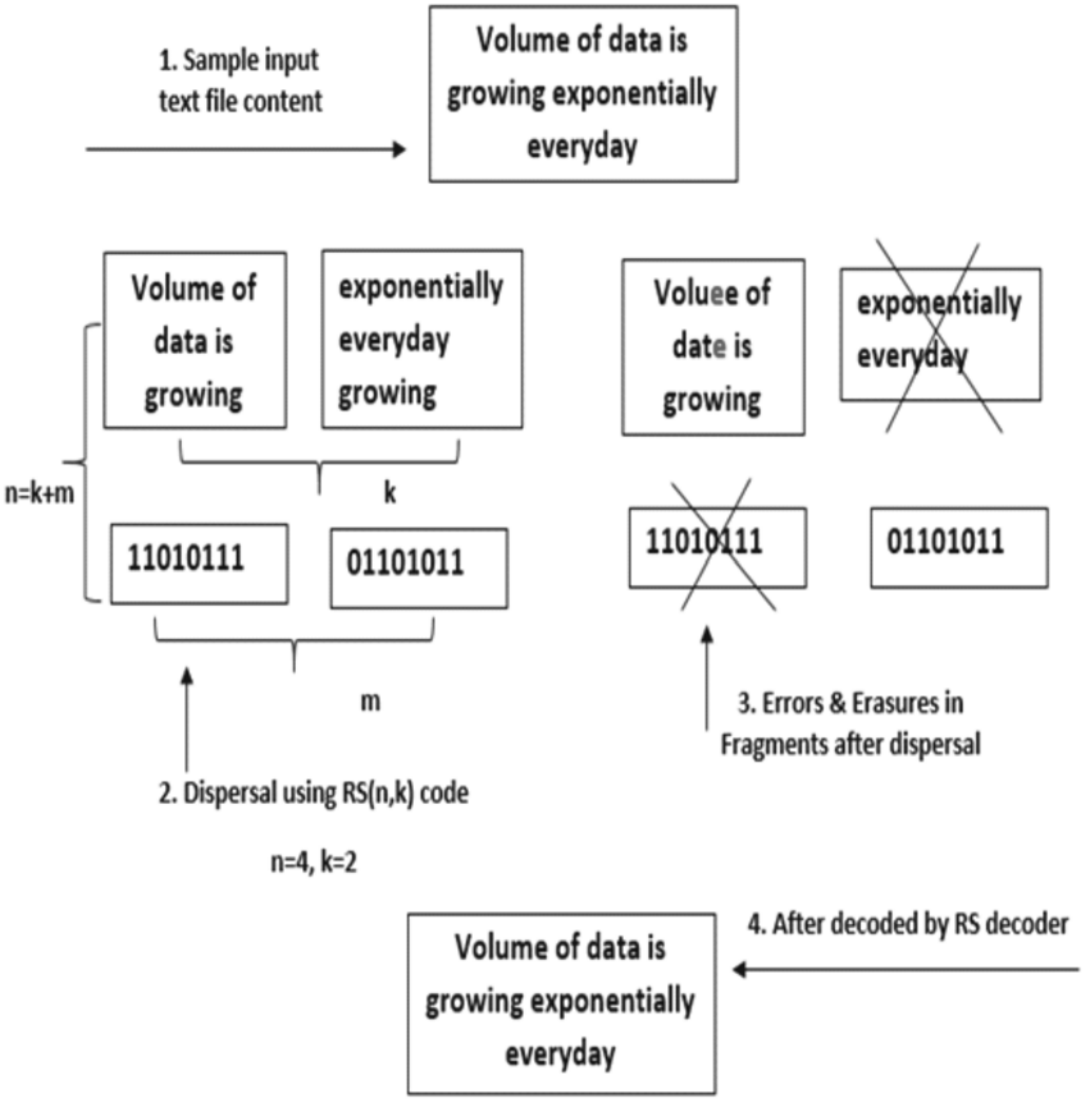

Fig. 6. Dispersal using RS Codes.

5. DQ-D WITH IDA

IDA is a method to split the file *f* into *n* pieces in such a way that it can be reconstructed from subset of *k* pieces where, $k < n$. Data can be retrieved if *k* pieces out of *n* is obtained [3] [9] [19]. Even $k-1$ will not able to reconstruct the data. It can reconstruct the data even if *n-k* pieces are lost but if there is an erroneous data it will be reconstructed as such as illustrated in Fig.7

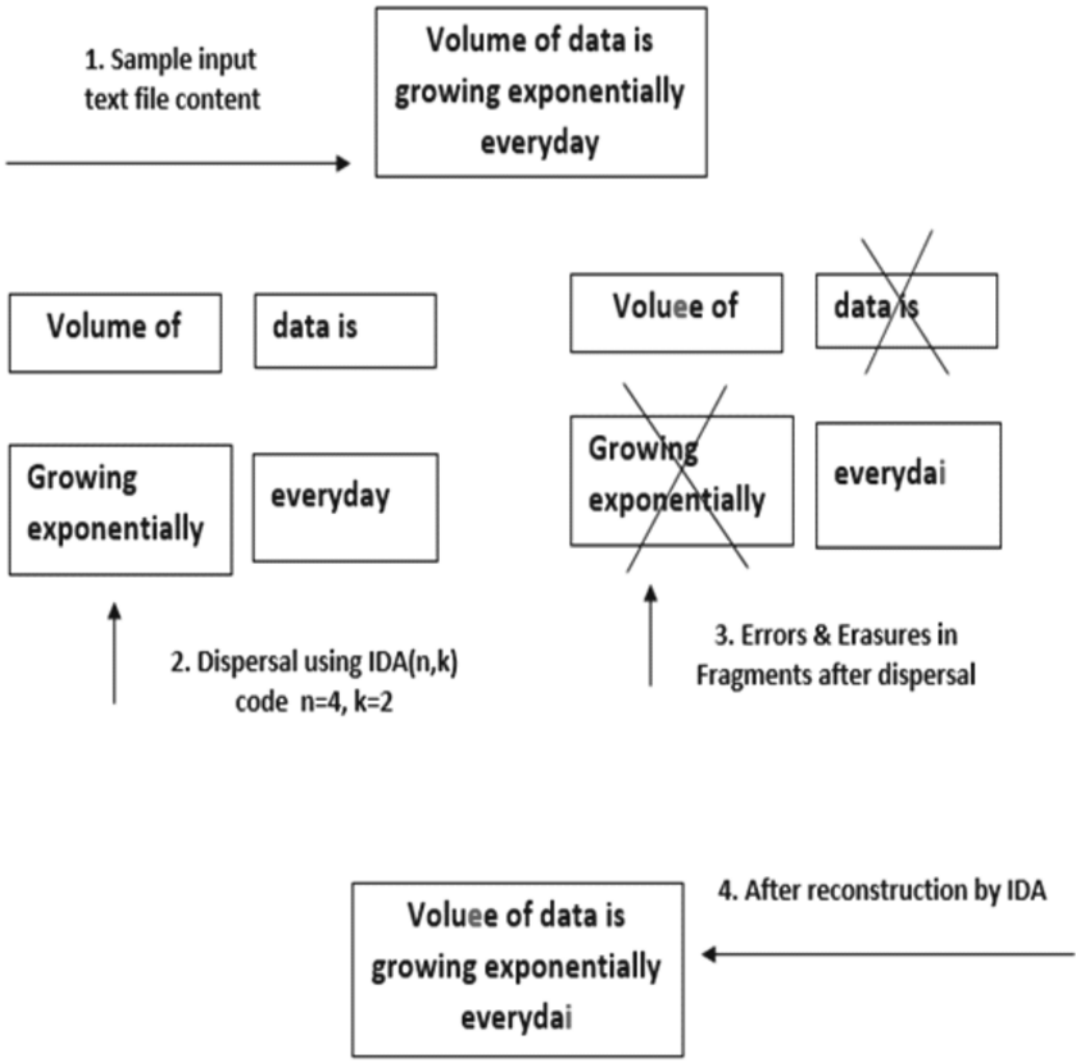

Fig. 7. Dispersal using IDA.

6. RESULTS AND ANALYSIS

To analyze the performance of DQ system in the presence of hybrid failures, a model for systematic injection of adversarial effect in DSS is designed and is discussed thoroughly in [20] Through Analytical and Experimental evaluations, the work compare features viz. Availability, Success rate (read & write), Response time, Fault tolerance capability (with Errors and Erasures) and security of dispersal technique incorporated with Dual Quorum (DQ-D) against the replication technique incorporated with Dual Quorum (DQ-R) in the presence of hybrid failure.

A. Availability

Availability is defined as as the number of client requests successfully processed by the system over the total number of requests submitted to the system during a given period of time. [2] When we incorporate dispersal technique with DQ protocol, the read request needs access to *k* servers to perform a successful read operation and write operation is performed over n servers. The request will be rejected by the system, if there are insufficient servers to process the request. The observation is made for various (*n*, *k*) combinations which are used by GFS, Microsoft Azure, Facebook RAID storage and its availability is shown in fig.8.

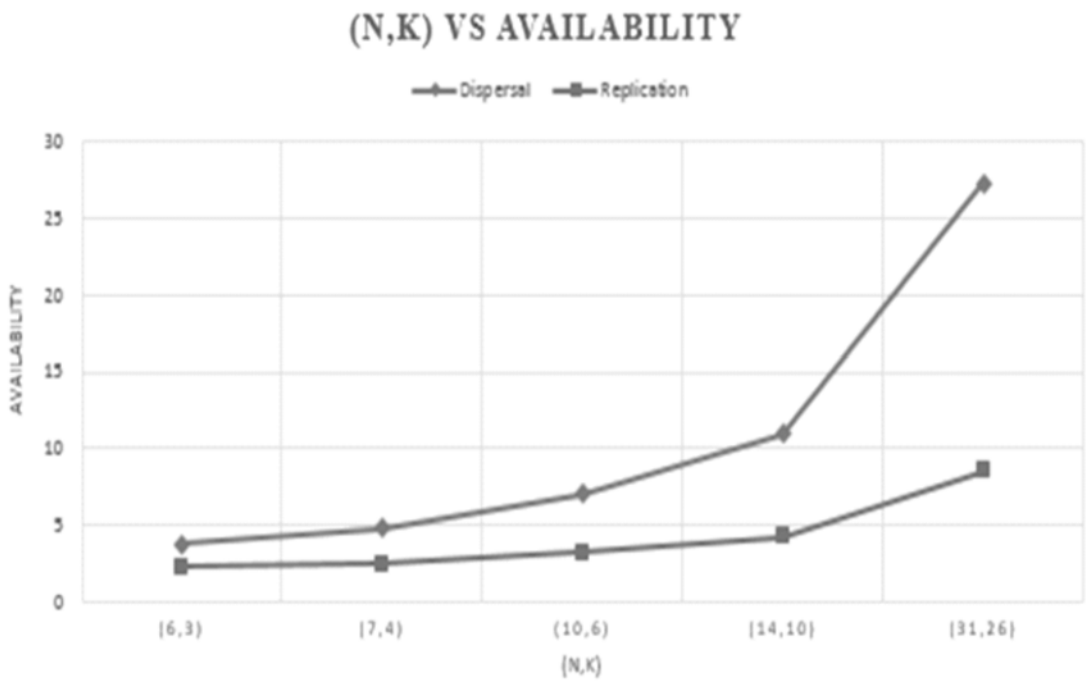

Fig. 8. Availability for various (*n*, *k*) Combination.

From Fig.8 it is clear that the availability of the system is increased with increase in *n* value and the dispersal technique employed in DQ is in par with the DQ employed with Replication technique.

B. Read with Hybrid Failures

The performance of the system in presence of hybrid failures is analyzed in this work. A comparison is made between the replication technique and dispersal technique (both IDA & RS Codes) as illustrated in Fig.9

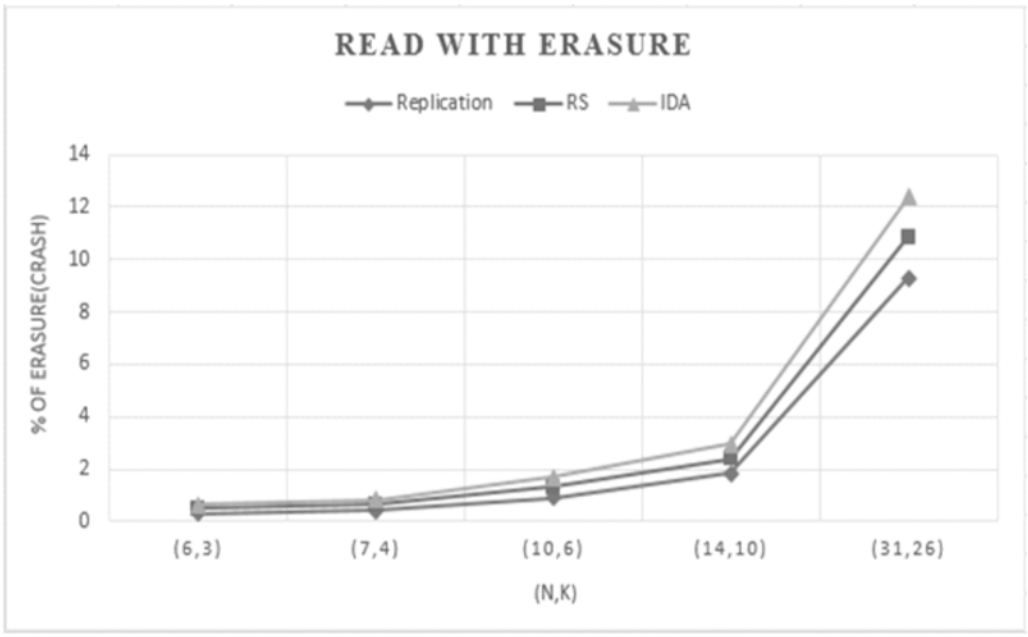

Fig. 9. Read with Erasure

It is observed that both the dispersal techniques, IDA and RS Codes are able to tolerate ($n - k$) erasure and are in par with each other whereas replication techniques are capable of tolerating ($n - 1$)erasure. Compared to replication, dispersal technique provides the best result as depicted in Fig.9.

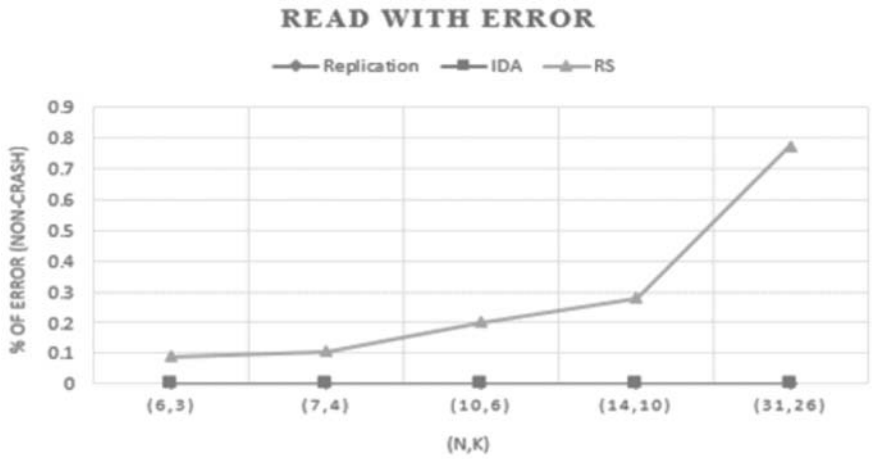

Fig. 10. Read with Error.

It is observed that the replication and IDA techniques does not have the ability to serve the client's request in the presence of Non-Crash faults and are equivalent in this case as in Fig.10.

C. Response Time

The time taken to respond for a read or write request is defined as response time [2]. We analyze the response time of Dual Quorum (DQ) similar to the evaluation done in DQ replication [2]. The response time of DQ with Dispersal is similar to that of DQ with replication technique as shown in Fig.11.

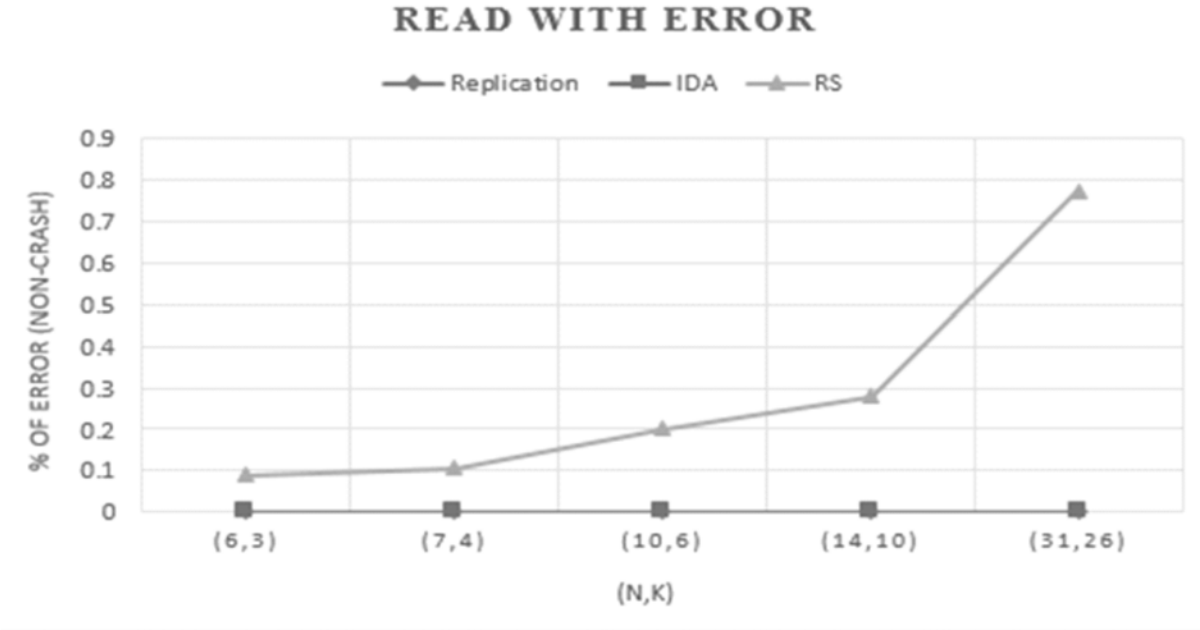

Fig. 11. Write ratio versus Response Time.

D. Security

Dual Quorum protocol when incorporated with Replication technique, it is easy to get access to the data because it creates exact copy of the data in all the replicas. So when an adversary gets access to any one of the replica, security is lost. On the other hand if DQ protocol is incorporated with dispersal technique, it requires access to *k* nodes which is difficult for an adversary to get the original data. Even if an adversary gets access to any of the nodes, he may get only part of the original data which is of no use. Compared to replication, using Dispersal technique with DQ protocol highly increases the security. All the above results showed that it is a best choice to use dispersal technique with DQ so that we can design an efficient Distributed Storage Systems.

7. CONCLUSION

Apart from the primary needs such as availability, consistency and fault tolerance, security, confidentiality in particular has significant relevance in designing a reliable DSS. When the storage environment is distributed, there is ampleopportunities for various threats in the system; this motivated us to consider hybrid failures in our work. The paper models hybrid failures to analyze the performance of DSS in presence of adversaries. The model enables systematic injection of failures at DSS. The results and observations highlight the applicability of an integrated DQ with dispersal to obtain good performance in the presence of failures.

ACKNOWLEDGMENT

I would like to thank my Guide Dr. C.K.Shyamala, Assistant Professor, Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Coimbatore for her guidance, reviews and valuable comments which helped a lot in writing this paper.

REFERENCES

1. Bowers KD, Juels A, Oprea A. HAIL: a high-availability and integrity layer for cloud storage. InProceedings of the 16th ACM conference on Computer and communications security 2009 Nov 9 (pp. 187-198). ACM.
2. Gao L, Dahlin M, Zheng J, Alvisi L, Iyengar A. Dual-Quorum: A Highly Available and Consistent Replication System for Edge Services. Dependable and Secure Computing, IEEE Transactions on. 2010 Apr;7(2):159-74.
3. Nirmala SJ, Bhanu SM, Patel AA. A Comparative study of the secret sharing algorithms for secure data in the cloud. International Journal on Cloud Computing: Services and Architecture (IJCCSA). 2012 Aug;2(4):63-71.
4. Martin KM. Challenging the adversary model in secret sharing schemes. Coding and Cryptography II, Proceedings of the Royal Flemish Academy of Belgium for Science and the Arts. 2008:45-63.
5. Bal HE, Kaashoek MF, Tanenbaum AS, Jansen J. Replication techniques for speeding up parallel applications on distributed systems. Concurrency: Practice and Experience. 1992 Aug 1;4(5):337-55.
6. Rabin MO. Efficient dispersal of information for security, load balancing, and fault tolerance. Journal of the ACM (JACM). 1989 Apr 1;36(2):335-48.
7. Kumar S, Gupta R. Bit error rate analysis of Reed-Solomon code for efficient communication system. International Journal of Computer Applications. 2011 Sep;30(12):11-5.
8. Shyamala CK, Vidya NV. Structuring Reliable Distributed Storages. InIntelligent Systems Technologies and Applications 2016 (pp. 235-245). Springer International Publishing.
9. Agrawal D, El Abbadi A. Integrating security with fault-tolerant distributed databases. The Computer Journal. 1990 Jan 1;33(1):71-8.
10. Peleg D, Wool A. The availability of quorum systems. Information and Computation. 1995 Dec 31;123(2):210-23.
11. Abdallah A, Salleh M. Analysis and Comparison the Security and Performance of Secret Sharing Schemes. Asian Journal of Information Technology. 2015;14(2):74-83.
12. Shyamala CK, Vidya NV. Dual Quorum-Dispersal, A Novel Approach for Reliable and Consistent ECC based storages.International Journal of Applied Engineering and Research (IJAER).2015;10(3): 6901-6917.
13. Debains C, Alvarez-Tabio P, Zhao D, Raicu I. IStore: Towards High Efficiency, Performance, and Reliability in Distributed Data Storage with Information Dispersal Algorithms. under review at IEEE MSST. 2013.
14. Bose R. Information theory, coding and cryptography. Tata McGraw-Hill Education; 2008.
15. Bansal S, Sharma S, Trivedi I. A detailed review of fault-tolerance techniques in Distributed System. International Journal on Internet and Distributed Computing Systems. 2011 Jun 1;1(1).
16. Shyamala CK, Sharanya RR. STUDY OF ADVERSARY MODELS IN DISTRIBUTED STORAGE SYSTEMS. Global Journal of Pure and Applied Mathematics (GJPAM).;11(1):2015.

17. Ling J, Jiang X. Distributed storage method based on information dispersal algorithm. InInstrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on 2013 Dec 23 (pp. 624-626). IEEE.

18. Cachin C, Tessaro S. Asynchronous verifiable information dispersal. InReliable Distributed Systems, 2005. SRDS 2005. 24th IEEE Symposium on 2005 Oct 26 (pp. 191-201). IEEE.

19. Parakh A. New Information Dispersal Techniques for Trustworthy Computing(Doctoral dissertation, Oklahoma State University).

20. Shyamala CK, Sharanya RR. Adversarial Effect in Distributed Storage Systems. In Indian Journal of Science and Technology(IJST), 2016 ICIIECS'16 (Accepted)

21. Rajendra AB, Sheshadri HS Visual Secret Sharing Biometric Authentication using steganography, 2014 june 3 (pp. 1021-1026).