World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:1, 2016

A Hybrid P2P Storage Scheme Based on Erasure Coding and Replication

Usman Mahmood, Khawaja M. U. Suleman

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10004046.pdf

Abstract**—**A peer-to-peer storage system has challenges like; peer availability, data protection, churn rate. To address these challenges different redundancy, replacement and repair schemes are used. This paper presents a hybrid scheme of redundancy using replication and erasure coding. We calculate and compare the storage, access, and maintenance costs of our proposed scheme with existing redundancy schemes. For realistic behaviour of peers a trace of live peer-to-peer system is used. The effect of different replication, and repair schemes are also shown. The proposed hybrid scheme performs better than existing double coding hybrid scheme in all metrics and have an improved maintenance cost than hierarchical codes.

Keywords**—**Erasure Coding, P2P, Redundancy, Replication.

I. Introduction

INITIAL work done in the area of peer-to-peer networks was related to file sharing which led to the creation of other applications related to distributed storage and Voice-over-Ip. The main issue of any peer-to-peer system is of unreliable peers and churn – joining and leaving of peers. Since data availability is much more important concept in case of a peer-to-peer storage system. The issue of unreliable peers and churn is much more important to address in a peer-to-peer storage system, as compared to other peer-to-peer systems. To solve this issue, data redundancy is used. Two main schemes of data redundancy are replication and erasure coding. Replication simply copies an object onto multiple peers and needs only one of the peers to be online to access the object. On the other hand, erasure coding breaks a single object, such as a file, into m chunks; then, an additional $n - m$ parity objects are added to it to make it a total of n chunks. Erasure coding must retrieve data from at least m different peers out of n to re-construct the object. One simple hybrid scheme presented in [1], [2] uses replication to replicate the original file on one peer along with erasure coded chunks on other peers. The second hybrid scheme presented in [6] makes two copies of every erasure coded chunk to make a total of 2n chunks.

Studies like [3]-[5] compare the two schemes theoretically and claim that erasure coding is more feasible but [6] shows that erasure coding has its own disadvantages in terms of maintenance cost (Bandwidth cost to maintain the system/availability). References [1], [2] compare hybrid scheme and regenerating codes and the results show that regenerating codes perform better than the simple hybrid scheme only with a stable environment with very high peers' availability. A stable environment with high peer availability is not a common case in terms of p2p storage application.

In this work, we compare replication, erasure coding and hybrid schemes using live trace of a peer-to-peer system on the basis of storage, maintenance and access cost. The effect of different replacement and repair schemes are also presented. Further we compare our proposed scheme with existing hybrid schemes. The results of hierarchal codes [7] are compared with the results of our proposed scheme.

Since, a simulation of peer availability does not show the accurate results because of the random behaviour of peers. So, the experiments are performed on traces of a live peer-to-peer network.

The paper is organised as follows: Section II describes the related work, Section III explains the redundancy schemes used, Section IV gives the results and Section V explains the conclusion.

II. Related Work

A peer-to-peer storage system is greatly affected by the redundancy, replacement and repair schemes used. These schemes describe how to make data redundant and maintain the redundancy. Data redundancy is a very important issue in the field of storage systems. There are mainly two ways of doing data redundancy, replication and erasure coding. Replication is the simplest of the technique which simply a file onto multiple locations. Many p2p storage systems [8] – [10] use replication as redundancy scheme. Different variations of replication exist which replicate a file completely or in chunks. Reference [11] uses erasure coding to make data redundant. Replacement and repair schemes controls how and when to repair the redundant data. These schemes are evaluated using trace of p2p systems as done in [1], [6] or by simulating a model of peers as done in [2]. Studies like [3]-[5] compare replication and erasure coding on the basis of data availability and storage cost. References [1], [2] compare redundancy schemes on maintenance cost as well. Williams et al. [12] also uses access cost as a metric to compare redundancy schemes.

Redundancy schemes can be classified into two categories; replication and erasure coding. Replication is fairly simple but erasure coding can be done using different techniques, the main schemes are: 1) MDS codes, 2) LDPC Codes, 3) RC codes and 4) Hierarchal codes. Maximum Distance Separable (MDS) are the codes which allow (n) fault tolerant nodes where (n) is the number of coding nodes. Example of MDS codes are EVENODD [13] and X-Code [14], in these schemes

Usman Mahmood is with the National University of Computer and Emerging Sciences, Pakistan (e-mail: usman.mahmood134@gmail.com).

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10004046.pdf

k is set to be equal to 2. For values of k greater than 2, erasure coding on Galois Fields are used and is the basis of Reed-Solomon codes [15]. There are many implementations of Reed-Solomon codes such as [16]-[18] which provide parity objects greater than 2 but they have a cost in terms of high computational complexity. To decrease the complexity of generating codes, Tornado Codes [19] and LT Codes [20] are proposed by Luby et al., these techniques are the type of LPDC codes. Studies like [1], [2] have used hybrid scheme which uses both erasure coding and replication to have the pros of both schemes. Reference [1] gives a new coding technique, named Regenerating codes and claims that RC proves to better in terms on maintenance cost in a very stable environment but performs slightly worse than simple hybrid scheme in an unstable environment with low peers' availability. Reference [2] proves that hybrid schemes performs as good as Regenerating Codes. Reference [7] presents hierarchal codes and proves to be better than standard Reed Solomon codes in terms of maintenance.

III. Redundancy Schemes

Replication is the simplest redundancy scheme in which data is replicated completely or partially on multiple peers. *Erasure Coding*, creates m chunks by partitioning the original data and then creates k parity chunks to have a total of n (m+k) chunks, data can be recovered using any of the m chunks from n. Erasure coding has high maintenance cost that is why many researchers have used and compared *Hybrid scheme*. In simple hybrid scheme [1], [2] erasure coding and replication are combined by replicating the complete file on a single peer (primary copy) and then using erasure coding to create chunks of data including both data chunks and parity chunks and transferring them onto to other peers. Fig. 1 shows the architecture of this scheme.

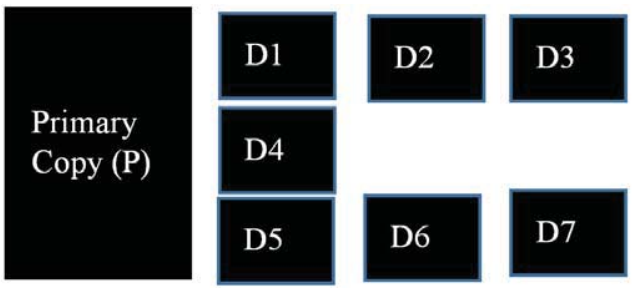

Fig. 1 Architecture of Hybrid Scheme

The double coding scheme presented in [2] stores each of the n (m+k) chunks on two different nodes so it needs a total of 2n peers to store the data. Simple hybrid scheme has security concern as complete data is stored on an ordinary peer, whereas double coding scheme seems to have high storage cost. The 2Rhybrid scheme proposed in this paper also replicates the n chunks created by erasure coding but not using n additional peers but replicates them on only 2 peers. 2Rhybrid scheme requires n+2 peers. The concept of this scheme is when you have 'n' data chunks taken from erasure coding, transfer half of them on to a single peer (P1) and the other half on peer (P2). This way no peer has the complete data, they do not have part of the complete data. P1 and P2 would have only erasure coded chunks, which cannot be transformed into the actual data until they have 'm' chunks. This schemes works when m >(m+k)/2, which is the usual case. Fig. 2 demonstrates the architecture of this scheme.

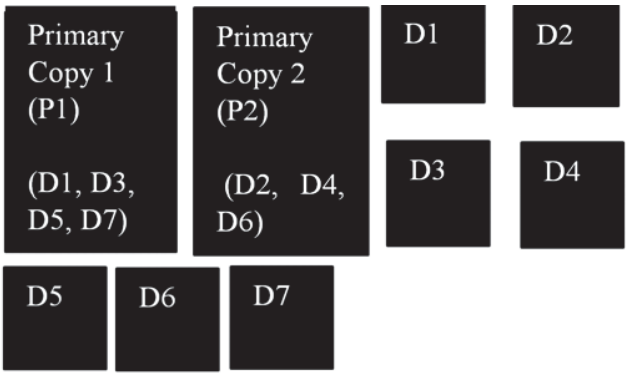

Fig. 2 Architecture of 2RHybrid: Each block represents a node

Decoding of erasure coding requires data from any 'm' chunks, without hybrid scheme 'm' chunks are stored on 'm' different peers. We have used two lazy schemes to repair data, one is time based which repairs a peer if a peer stays dead for an hour. The other scheme repairs have a fixed number of peers storing same file go offline. In hybrid scheme if the peer storing the complete file is online then the complete data is recovered from there, and parts of data are sent to peers who have lost the data. In double coding, if a peer (b) needs to be repaired then the other peer which stores the replica of the data on peer (b) is used to repair that data. If two peers storing the same chunk of a file go offline, then data from m peers have to be accessed. In 2Rhybrid scheme, if a peer has to be repaired, the peer storing the relevant primary copy is pinged, if that is available then the data is accessed from there, if not then complete data is accessed using other primary copy if it is available, otherwise m peers have to be pinged to get complete data.

IV. Experimental Work

A simulated model of peers cannot exhibit the random behavior of peers so we have used KAD [21] trace of peers. An important metric that is used throughout the thesis is *stretch.* Stretch is defined as the total data stored on the peer-to-peer network against the total data of user. For example, if every data of user is replicated twice then the stretch value is 2.

$$\text{stretch} = \text{total Data Stored} / \text{total user data}$$

In this paper, the data availability is set to 99.99% and corresponding stretch, churn cost and access cost are computed using different redundancy, replacement and repair schemes.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10004046.pdf

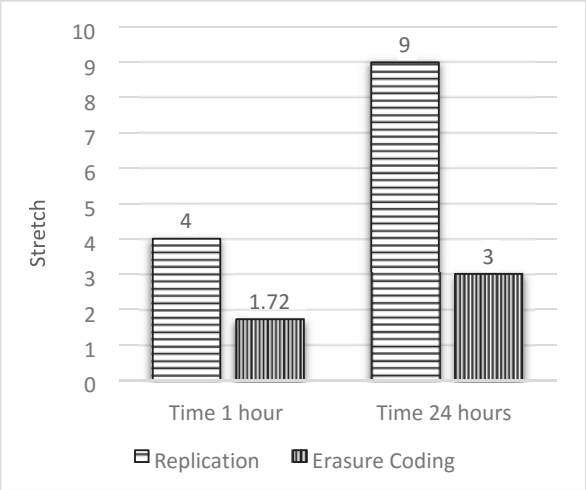

Fig. 3 Stretch required for replication and erasure coding

A. Stretch Required

The appropriate stretch level required for both replication and erasure coding is shown in Fig. 3. The settings for both the schemes are described below.

1. Replication

Replication scheme is first used for data redundancy and the appropriate replication is found for different time out schemes. We have used a lazy repair policy (time based), which replaces a peer with another peer with a random peer when a peer is offline for more than an hour, results are also calculated for the case when a peer remains offline for consecutive 24 hours. Time hour of 1 is fairly strict, as peers keep leaving and joining the system. Whereas, time hour of 24 hours seems more realistic in a peer-to-peer network. Both settings have their own advantages and disadvantages. Lower value of time in Fig. 3 results in smaller stretch value but increases churn cost as more peers are replaced than higher time hour. Peers are replaced randomly in both cases. The stretch level shown in Fig. 3 is for data availability of 99.99%.

2. Erasure Coding:

For erasure coding, each data file is divided into 7 chunks and additional parity chunks are then added to make total of n chunks. So, for the results 'm' was set to 7. For time interval of 1 hour, the appropriate stretch value is 1.72. It means that for every 1 unit of user data we have to store 1.72 unit of data. The unit can be a bit, byte or terabyte. For m =7 n would be equal to 1.72*7 = 12. Each data file is partitioned into 7 chunks and then an additional 5 parity chunks are added to make it a total of 12 chunks.

Erasure coding is surely a better technique of data redundancy as it requires a significant less amount of stretch to give same level of data availability.

B. Churn Cost

In a peer-to-peer network, churn is defined as the joining and leaving of peers in the network. In the experimental work churn cost is defined as the percentage of total user data transferred per hour to ensure the availability of 99.99%. When a peer storing some data leaves the network that data has to be transferred to other peers to ensure the availability of data. The continuous joining and leaving of peers cost both in computational (in case of erasure coding) and bandwidth required to recreate data to transfer it to new peers. Fig. 4 shows the churn cost of replication, erasure coding, SHS, double coding and 2RH schemes with data availability set to 99.99% and time out of 1 hour is used as repair scheme and random replication is done for the dead peers.

Churn cost of 0.1 means that 10% of user data is transferred per hour in the network to maintain the required availability. It is clearly visible that churn cost of erasure coding is much greater than replication, the reason for that is that erasure coding stores a piece of data on more peers as compared to replication. In this instance, replication stored each file on 4 peers, and erasure coding stored each file on 12 peers.

Whenever a peer goes offline in case of replication or in case of erasure coding, the amount of data transferred is same, since erasure coding needs complete data to create a new chunk. As erasure coding stores a file on 12 peers, so more replacements had to be made in this scenario, than replication which stored a file on 4 peers. The churn cost of hybrid scheme (SHS) is much less than erasure coding and is even less than replication, the reason of that is that only 1 peer stores the complete file, so the complete file has to be transferred only when the peer storing the complete file is to be replaced, which in this case is only 1 peer as compared to 4 in replication and 12 in erasure coding. As long as the peer storing the complete file is available, churn cost of only 1/7 (0.14) has to be paid for the rest of peers.

SHS has a great issue of security as one peer has the complete data. Because of this reason we have used 2RH scheme. In 2RHybrid scheme stretch remains same as SHS because in 2RHybrid scheme the availability gets over 99.99% with the same value of n; therefore, n was reduced to 9 in this case. The churn cost of 2Rhybrid scheme is a little greater of simple hybrid scheme. The reason of that is that the complete data is now divided onto two peers. This can be explained using an example. Referring to Fig. 2, if d1 and d2 both goes offline, they now need both p1 and p2 to be online to have a churn cost of 0.142 each. As the probability of p1 and p2 being alone together is less than only p1 being online so the churn cost increases. The result shows that 2RH scheme is a better scheme than double coding in terms on churn cost and stretch, the reason of this is that in double coding, complete data has to be transferred on ever pair storing the same chunk getting offline and there are 11 pairs. In 2RH scheme complete data has to be transferred in only one case when both primary copies go offline. 2RHbyrid gives more than 70% improvement over Erasure Coding in maintenance which is much better than the 19% improvement made by Hierarchal codes presented in [7]. It gives an idea of the improvement made by our proposed scheme.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10004046.pdf

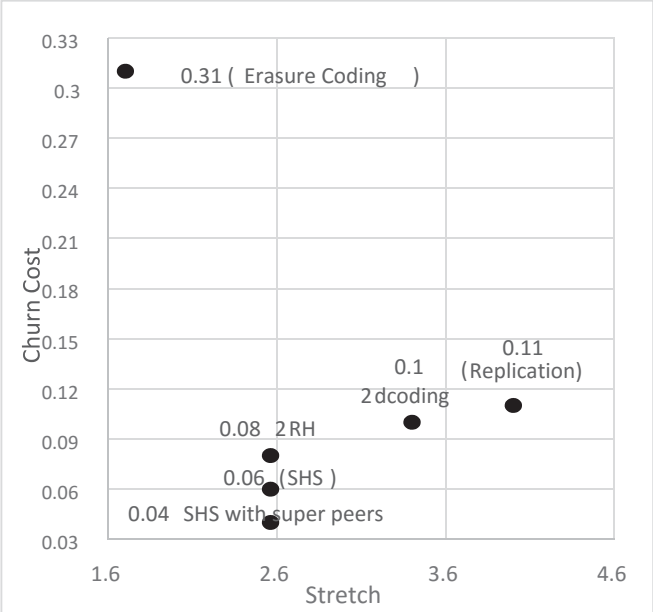

Fig. 4 Churn Cost of All Schemes

C. Access Cost

Access cost is another important feature in a peer-to-peer network, it defines the cost of accessing a data piece. Access cost is the number of pings required before the complete data can be fetched. Since in replication the complete file is placed at a peer, so mostly only 1 ping is required to get the data. Whereas, in erasure coding a peer stores only a part of the complete file and hence at least m pings are required to access the peers and to access the data. In this experiment the value of m is 7. So, in a scenario where peers' availability is 100 % and m is set to 7, replication require has to access only 1 peer to fetch the data whereas, erasure coding requires to access 7 peers to get the complete data. Therefore, in Fig. 5, it can be clearly seen that erasure coding has to access much more peers than replication.

The access cost of hybrid scheme is also much less than erasure coding, because if the peer storing the complete file is available then access cost of only 1 had to be paid. The access cost of hybrid is a little greater than replication because if the peer storing the complete file was not online then at least 7 peers had to be accessed. But since, time out scheme of 1 hour was used so that peer was mostly online. The access cost of 2Rhybrid scheme is greater than simple hybrid scheme, because the minimum access cost is now 2 instead of 1. Also both peers being online at the same time is less likely than only one being online. In Fig. 5, it is clear that 2RH scheme is much better than 2dcoding as in 2dcoding in ideal situation min access cost is m, as m different peers have to be accessed whereas in ideal situation, churn cost of 2RH is 2.

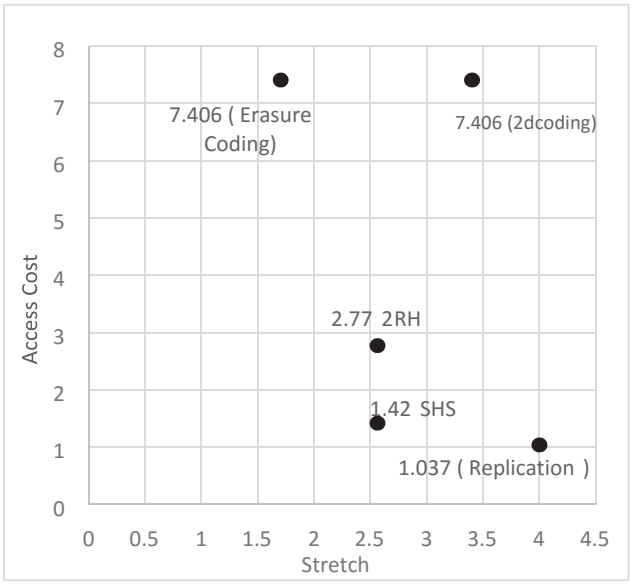

Fig. 5 Access Cost of All Schemes

Discussion

All the schemes mentioned above have their own pros and cons. 2RH scheme proves to be the best possible variation among all.

D. Replacement Scheme

All the experiments discussed above are performed using random replacement scheme, but there are many other techniques of replacement. One of those techniques 'highest up count' is used to see the effects on churn cost and access cost. In highest up count, after the random initial selection of core group, all the peers in the general group are pinged after every 5 minutes to see if they are alive or not.

If a peer is alive, it's up count is increased by 1. Replacement is made with the peer which has the highest up count in the general population and is online at that time. The effect on churn and access cost are discussed below.

1. Churn Cost

Highest up count replacement technique really shows promising results in case of churn cost, especially in erasure coding as it has more room for improvement. The main reason for such improvement is that the peers selected as replacement peers remain online for much longer time than other peers, so they go less offline so much less replacements has to be done. As number of replacements has direct effect on the churn cost so the decrease in number of replacements decreased the churn cost. The problem with this technique is that more data will be stored on good peers, as in a peer-to-peer storage system there has to be a limit on the amount of data stored on each peer. So, that limit has should be consider when using this scheme, but still the results would still be better than random replacement scheme.

2. Access Cost

The access cost also decreases but not as much as the churn cost. The reason of that is that there was not much room for

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10004046.pdf

improvement, erasure coding has a minimum access cost of 7, so the maximum improvement could only be 0.406 in case of erasure coding. Peers stay online for longer time, so the first 7 peers which were accessed by the system are mostly online so the access cost is much closer to 7 by using highest up count technique in erasure coding. Same effects can be seen in replication. Because of high peer availability in the case of Highest Up Count scheme for replacement, stretch can also be reduced. This effect is not calculated but the increased in data availability because of this scheme is show in the last section.

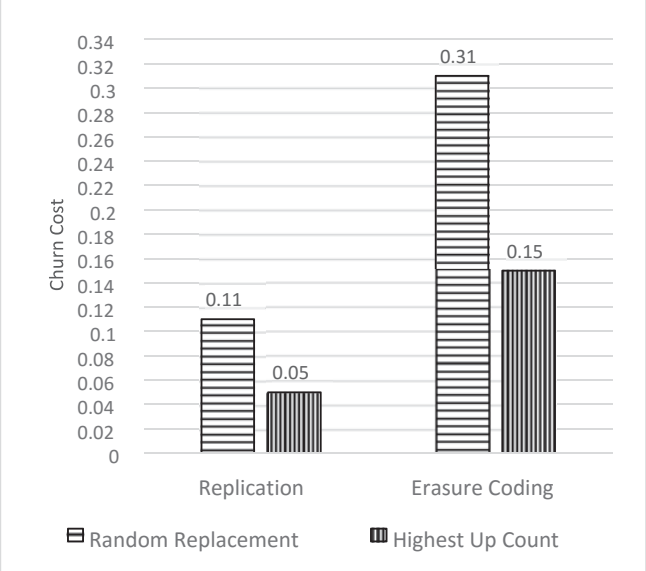

Fig. 6 Churn Cost of Replication & Erasure Coding with Random Replacement and Highest Up count replacement

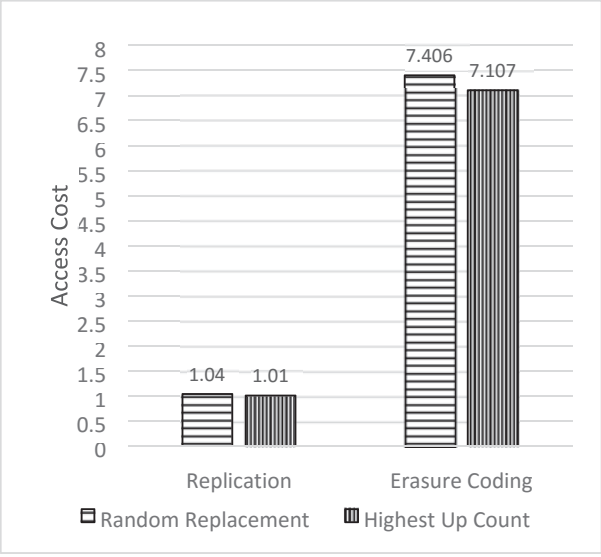

Fig. 7 Access Cost of Replication & Erasure Coding with Random Replacement and Highest Up count replacement

E. Repair Policy

To see the effect of the repair policy on churn and access cost another lazy approach is used. In this repair policy replacement is not done for replication until 50% of peers storing a file are not offline, and 20% in case of erasure coding. The effects are shown in Figs. 8, 9.

1. Churn Cost

In this technique, much less replacements are made because there are many cases where 1 or 2 peers go offline for an hour but come online again, if time out policy of hour is used then replication will occur and hence churn cost will increase. The current scheme does not perform replication until it is necessary.

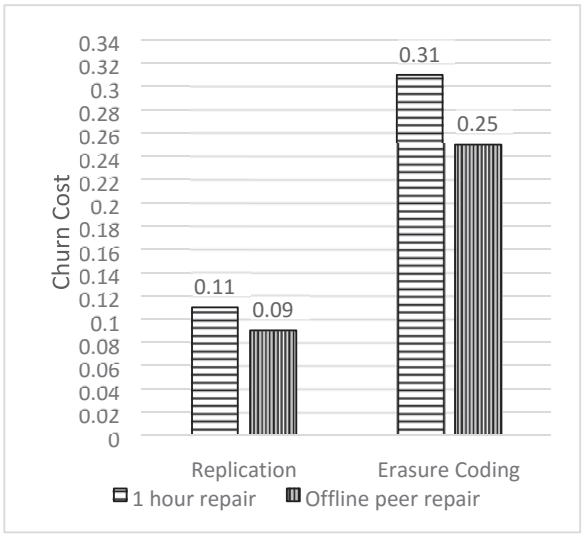

Fig. 8 Churn Cost of Replication & Erasure Coding with 1 Hour Time and Offline Peers

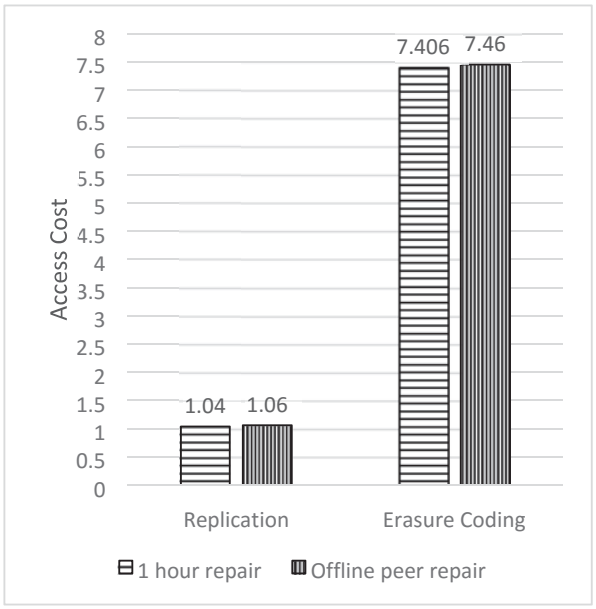

Fig. 9 Access Cost of Replication & Erasure Coding with 1 Hour Time and Offline Peers

2. Access Cost

The access cost increased a little, which is pretty understandable because there are much more cases in which

some of the peers storing a file are offline and they are not replaced. So, access cost does increase in this repair policy.

F. Data Availability

The better techniques of replacement and repair not only helps with access cost and churn cost but it also increases the data availability. Fig. 10 shows the increase in data availability in case of highest up count and repair done based on peer availability. The increased in availability shows that the stretch value can be reduced to get the required availability of 99.99%.

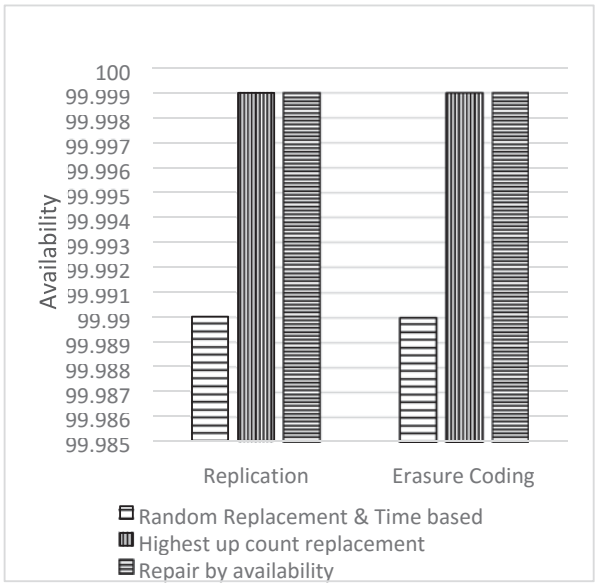

Fig. 10 Minimum Average Availability of Data

V. Conclusion

This work has endeavored to find the best way of applying redundancy and maintenance which would provide high data availability with low maintenance and access cost. Our work focused on different schemes of applying data redundancy and tested them on storage, maintenance and access cost. The results show that in an unstable peer-to-peer environment with low peer availability our proposed hybrid scheme proves to be better in every metric than double coding and less maintenance cost than hierarchal codes. The proposed hybrid scheme has less storage cost than erasure coding. 2RH scheme also solves the security issue of simple hybrid scheme. 2RH scheme proves to be more feasible choice than the previous redundancy schemes.

References

[1] A. Dimakis, P. Godfrey, M. Wainwright, AND K. Ramchandran: The benefits of network coding for peer-to-peer storage systems. In Workshop on Network Coding, Theory, and Applications, 2007.

[2] J. Araujo, F. Girore, and J. Monteiro: Hybrid Approaches for Distributed Storage. Proceedings of the 4th international conference on Data management in grid and peer-to-peer systems, 2011.

[3] R. Bhagwan, D. Moore, S. Savage, G. Voelker: Replication strategies for highly available peer-to-peer storage. Future Directions in Distributed Computing (FuDiCO), 2002.

[4] C. Blake, R. Rodrigues: High availability, scalable storage, dynamic peer networks: pick two. Proceedings of the 9th Workshop on Hot Topics in Operating Systems (HotOS-IX), Lihue, Hawaii, 2003.

[5] H. Weatherspoon and J. Kubiatowicz: Erasure coding vs. replication: A quantitative comparison. Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002), March 2002.

[6] R. Rodrigues, B. Liskov: High Availability in DHTs: Erasure Coding vs. Replication. 4th International Workshop on Peer-to-Peer Systems (IPTPS'05). Ithaca, New York, USA. February 2005.

[7] A. Duminuco and E. Biersack, "Hierarchical codes: How to make erasure codes attractive for peer-to-peer storage systems," in IEEE P2P, 2008.

[8] A. Adya, W. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. Douceur, J. Howell, J. Lorch, M. Theimer, R. Wattenhofer: FARSITE: federated, available, and reliable storage for an incompletely trusted environment. 5th Symposium on Operating Systems Design and Implementation (OSDI), 2002.

[9] P. Druschel and A. Rowstron. PAST: A large-scale, persistent peer-to-peer storage utility. In USENIX Workshop on Hot Topics in Operating Systems (HotOS), 2001.

[10] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative Storage with cfs. In ACM Symposium on Operating Systems Principles (SOSP), 2001.

[11] A. Haeberlen, A. Mislove, and P. Druschel. Glacier: highly durable, decentralized storage despite massive correlated failures. In USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2005.

[12] Williams, C., Huibonhoa, P., Holliday, J., Hospodor, A., Schwarz, T.: Redundancy management for P2P storage. In: Proc. of the Seventh IEEE Int. Symp. On Cluster Computing g and the Grid, CCGRID 2007, pp. 15–22. IEEE Computer Society, Washington, DC (2007).

[13] M. Blaum, J. Brady, J. Bruck, and J. Menon. EVENODD: An efficient scheme for tolerating double disk failures in RAID.

[14] L. Xu and J. Bruck. X-Code: MDS array codes with optimal encoding. *IEEE Transactions on Information Theory*, 45(1):272– 276, January 1999.

[15] I.S. Reed and G. Solomon: Polynomial Codes over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics, June 1960.

[16] F.J. MacWilliams Holland Publishing Company, Amsterdam, New *Applications*. IEEE Press, New York, 1994. and N.J.A. Sloane. *The Theory of Error- Correcting Codes, Part I*. North-

[17] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann. Practical loss-resilient codes. In *29th Annual ACM Symposium on Theory of Computing*, pages 150–159, El Paso, TX, 1997. ACM. York, Oxford, 1977.

[18] W. W. Peterson and E. J. Weldon, Jr. *ErrorCorrecting Codes, Second Edition*. The MIT Press, Cambridge, Massachusetts, 1972.

[19] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann. Practical loss-resilient codes. In ACM symposium on Theory of Computing (STOC), 1997.

[20] M. Luby. LT codes. In *IEEE Symposium on Foundations of Computer Science*, 2002.

[21] Moritz Steiner, Taoufik En-Najjary, and Ernst W. Biersack. A global view of KAD. Proc. of Internet Measurement Conference (IMC), October 2007, San Diego, USA.

Open Science Index, Computer and Information Engineering Vol:10, No:1, 2016 publications.waset.org/10004046.pdf