



# Adaptive Bandwidth-Efficient Recovery Techniques in Erasure-Coded Cloud Storage

Rekha Nachiappan<sup>(✉)</sup>, Bahman Javadi, Rodrigo N. Calheiros,  
and Kenan M. Matawie

School of Computing, Engineering and Mathematics, Western Sydney University,  
Sydney, Australia

{30045376,B.Javadi,R.Calheiros,K.Matawie}@westernsydney.edu.au

**Abstract.** In order to handle the dramatic growth of digital data, cloud storage systems demand novel techniques to improve data reliability. Replication and erasure codes are the most important data reliability techniques employed in cloud storage systems, but individually they have their own challenges. In this paper, we propose a hybrid technique employing proactive replication of data blocks in erasure-coded storage systems. The technique employs a set of erasure coding-agnostic bandwidth-efficient data recovery techniques that reduce the bandwidth used for recovery without compromising data reliability. Experiments show that our approach improves repair bandwidth efficiency and reduces network traffic in cloud storage systems with limited storage overhead compared to available recovery approaches.

## 1 Introduction

A recent trend in cloud storage systems is the adoption of erasure codes, as it provides excellent reliability with less storage overhead than replication [1]. For example, Facebook and Microsoft Azure replaced replication with erasure coding in parts of their data, resulting in significant cost savings in terms of storage overhead [2]. However, failure rates in large-scale cloud storage systems are high as such systems are composed of large number of hardware and software components. Repairing a single data block stored using Reed-Solomon( $n,k$ ) code requires  $k$  data blocks to be transferred over the network, while repairing a single data block in replication involves the transfer of one data block [3]. Hence, repair network traffic is increased by  $k$  times in Reed-Solomon( $n,k$ ) code compared to replication. The network traffic incurred by such data movement has also the extra drawback of increasing energy consumption significantly, resulting in extra costs for cloud service providers. Moreover, growing network traffic is regulated by network throttling, which affects read performance. All the above facts prevent cloud storage systems to adopt erasure codes in large scale.

Hardware failures (disk failures, machine failures, and latent sector errors) and temporary machine failures are the most common failures that affect durability and availability of data in cloud storage [2]. In order to avoid permanent

data loss due to hardware failures, contents in failed nodes or disks have to be restored in another hardware devices, a process that is known as *data recovery*. Data stored in a machine that experiences temporary outage will cause temporary data loss. Temporary data loss in erasure code is handled by degraded read, i.e., data blocks in the failed node are reconstructed and served using the next available  $k$  blocks. In order to avoid unnecessary repairs of short term transient node failures, data recovery is delayed for a certain amount of time. Google File System (GFS) delays recovery of unavailable nodes for 15 min. However, this affects availability and degrades read performance [5]. In contrast, when replication is used, degraded read is handled by simply redirecting the request to the next available replica.

As both replication and erasure coding have its own advantages, cloud storage systems require hybrid approaches in order to leverage the advantages of both methods, which are the recovery performance of replication and the storage efficiency of erasure coding. In this paper, we propose several novel recovery techniques. These techniques follow a proactive replication method. They replicates erasure-coded data blocks which are predicted to fail, keeping down repair network bandwidth/traffic at the same time without much overhead. We also showed that the ProDisk method proposed by Li et al. [13], reduces repair network bandwidth/traffic. All the aforementioned methods use machine and disk failure prediction techniques to predict hardware failures and long-time temporary machine outage. When hardware failures (permanent machine/disk failures) are predicted, proposed storage system immediately starts the recovery of data and proactively replicates erasure-coded data fragments in to permanent storage. When long-term machine failures are predicted, proposed storage system starts proactive recovery with the goal of maintaining data availability. During proactive recovery of long-term machine failures, data is written into dedicated temporary storage rather than on recovered blocks.

The amount of dedicated temporary storage required in the proposed approach is linearly related to the number of long term machine failures predicted over the time period. In order to address this issue, we introduce a novel method to proactively replicate hot data in temporary storage and apply lazy recovery for cold data. This reduces the recovery bandwidth/traffic significantly without increasing the temporary storage needed for supporting transient node failures.

## 2 Background and Motivation

In a distributed storage system, a data file is dispersed into multitude of interconnected nodes, which serves any end user request by tapping data from multiple nodes. Improving the resilience of distributed storage system with limited storage overhead is desirable. Replication is the simplest mean of increasing resiliency of the distributed storage. In replication, a data file is divided into multiple data blocks which are replicated into several locations such that failure of any data block in one location enables the user to access it from different location. However, reliability is directly proportional to storage overhead in replication.

Erasure coding is an important option to increase reliability with less storage overhead. In erasure coding, data file is divided into  $k$  data blocks and dispersed into  $n$  locations while adding  $n-k$  parity blocks. Upon any failure, a data block is reconstructed by downloading any  $k$  available data blocks. The data recovery in erasure coding increases recovery network bandwidth  $k$  times, compared to replication.

Facebook employed Reed-Solomon to only 8% of data in 3000 node production cluster and it has been estimated that if 50% of data were replaced with Reed-Solomon, repair network traffic would saturate their network links [4]. Increased repair network traffic is one of the major bottleneck to erasure coding becoming more pervasive in cloud storage systems. Novel blockchain-based cloud storage systems like Sia<sup>1</sup> and Storj<sup>2</sup> use consumer storage to serve their customer's storage needs. They suggest, as a means to improve reliability, the use of Reed-Solomon (60, 40) code. This means that, to reconstruct any missing data, 40 surviving data fragments have to be transferred to reconstruct any single failed data fragment. These novel storage systems demand more bandwidth-efficient recovery, which is the focus of this paper. The proactive recovery techniques proposed in this paper use several failure prediction methods. As these systems are running on end-users client, it may not be possible to apply existing hardware failures prediction techniques on the users computers. However, it is possible to predict the availability of user computers using availability logs. Hence it is possible to apply the proposed methods in blockchain-based cloud storage systems.

The main contribution of this paper is the definition of bandwidth-efficient recovery techniques based on client's needs without significant increase of permanent storage.

### 3 Related Work

A substantial amount of research concentrated on reducing repair bandwidth of erasure codes. Dimakis et al. [6] presented a theoretical framework for regeneration codes that can optimize recovery bandwidth for a given storage. However, exact repair of regeneration codes, matching information theoretic bound, remained unresolved. Following this, several works [2] showed that exact repair is possible for some parameters. Sathiamoorthy et al. [4], proposed Xorbas which reduces network traffic by half compared to Reed-Solomon codes with 14% additional storage overhead [4]. LRC in Windows Azure storage reduces repair network bandwidth significantly with the help of local parities, which have the side effect of increasing storage overhead by 1.33x compared to Reed-Solomon [1]. Hitchhiker code, built on top of Reed-Solomon code using "piggybacking" framework, reduces network traffic by 35% with some encoding time overhead incurred [7].

<sup>1</sup> <https://sia.tech/>.

<sup>2</sup> <https://storj.io/>.

Failure predictions in cloud storage systems offer cloud service providers an efficient proactive failure management in cloud storage. Various statistical and machine learning methods are used to predict failures in cloud storage systems. A few methods [8,9] are used to predict hard drive failures based on SMART attributes. Li et al. [9], achieved 95% predictions with False Alarm rate less than 0.1%. Many researches focused on predicting failures in distributed systems based on system logs. Javadi et al. [10], presented failure model as a predictive method of distributed systems availability and unavailability. Agrawal et al. [11], uses log messages to predict failures in Hadoop clusters.

Silberstein et al. [12], proposed lazy recovery to reduce recovery bandwidth in distributed storage by reducing the recovery rate. It reduces recovery bandwidth up to 76% compared to Reed-Solomon. However, applying this method on cloud storage affects read performance and data durability. Li et al. [13], used failure prediction techniques to implement proactive replication in erasure codes for reducing degraded read latency and improving read performance. Li et al. [14], defined a cost effective data reliability management mechanism to ensure reliability of massive data with minimum replication based on a generalized data reliability model. Wu et al. [15,16], used prediction tools to identify the upcoming events and proactively migrates the data blocks on the degraded device belonging to the hot data zones in the large-scale data centers.

## 4 The Proposed Cloud Storage System

The target system in this paper is an object storage that initially stores data with any appropriate erasure code to reduce storage overhead while maintaining reliability. Consider a distributed cloud storage system composed of a number of disks accommodated in a machine, group of machines in a rack, and several racks in a distributed storage. Data blocks stored in a disk can be determined as an *at-risk block* based on the machine and disks health status where it is stored. Machine and disk failure prediction algorithms run individually to predict disk/machine failure and machine unavailability. Since rack failures are transitory, the health of data blocks is determined with machine and disks health status. Data blocks that are marked as at-risk in this system are proactively replicated before the occurrence of failure based on the client's Service Level Agreement (SLA). Proactive replication reduces the number of blocks required for reconstructions in erasure coded cloud storage system. Hence, the system reduces network traffic with less storage overhead. This system utilizes various recovery schemes to reduce reconstruction bandwidth in erasure coded cloud storage systems.

### 4.1 Architecture and Design

An overview of the system architecture is depicted in Fig. 1. It is implemented as an extension of a regular object storage. Object storage manages data as objects where each object has both data and metadata. A dedicated proxy server

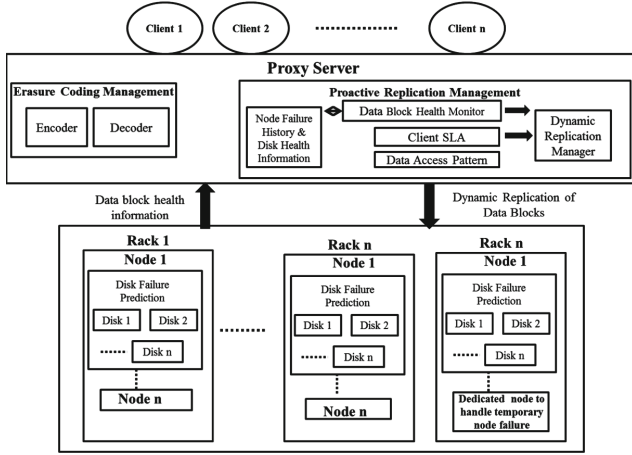


Fig. 1. Architecture of the proposed recovery techniques.

extends the support of encoding and decoding erasure codes. It also handles failures in storage systems. The object server stores and retrieves object data. Object server’s availability status and disks health status are reported to the proxy server, which is responsible for increasing or decreasing the data object’s replication factor. The system adjusts the replication factor of erasure coded objects when failures are predicted. The components of the architecture are discussed as follows.

**Disk Failure Prediction.** This module monitors the health status of individual disks and reports prediction results to the Node Failure History & Disk Health Information module in the proxy server. SMART is implemented on disks and it monitors, compares disk attributes and issues warnings. This SMART attributes are used to predict disk health status using various statistical and machine learning techniques [8,9]. Disk failures are calculated using classification and regression trees methods here [9].

**Proactive Replication Management.** Redundancy of data blocks are adjusted according to node/disk health status and client SLA.

*Node Failure History and Disk Health Information.* This module collects the information of disk health status and node failure history. Various statistical and machine learning techniques can be used to predict node’s Mean Time To Failure (MTTF) and Mean Time To Repair (MTTR). Based on node’s predicted MTTF and MTTR, node failures are classified as permanent, long time, or short time failures. Node’s MTTF and MTTR are calculated using various statistics of availability and unavailability [10].

*Data Block Health Monitor and Client SLA.* Failure predicted nodes and disks information are collected from Node Failure History and Disk Health Information module. It identifies the disks that are predicted to fail in the underlying storage system. It also identifies permanent, long term, and short term machine failures by predicting machines MTTF and MTTR. Permanent machine failures are handled as disk failures. This module sends failure information to the Dynamic Replication module, which takes an action when necessary. Clients can request various recovery schemes based on their needs. The client can define several reconstruction requests as follows,

- High durability, normal availability (ProDisk).
- High durability, high availability (ProMachine).
- High durability, high availability for hot and normal availability for cold data (ProHot).
- High durability, high availability for hot and low availability for cold data (ProHot\_LazyCold).

Based on the client SLA, the variable for different recovery scheme will be set.

*Data Access Pattern.* Data access patterns in a distributed storage can be used to identify the popularity of data blocks in real-time over a certain period of time. Based on their popularity, data blocks can be classified as hot, warm, or cold. As the access pattern changes, popularity of data blocks need to be updated. Various researches used popularity-based classification to improve durability, availability, and read performance of cloud storage systems [17]. Our approach combines both failure prediction and data access patterns to make the decisions. Data access pattern is used here to define hot data. We assume that data blocks with high access frequency have more chance to be accessed in the future and those are defined as hot. This module uses data access pattern to classified a block as hot data block and recorded as  $H = \{b_1, b_2, \dots\}$  where the block  $b_i$  is identified as hot.

*Dynamic Replication Manager.* This module collects information from Data Block Health Monitor, Client SLA, and Data Access Pattern module and activates various proposed recovery schemes, as follows:

- **ProDisk:** When disk failures/permanent machine failures are predicted, all the data blocks in the failure predicted disks (all disks in failure predicted machine) are proactively replicated permanently as described in [13]. In the occurrence of failure, the reference is made to the proactively replicated data instead of the typical reconstruction of erasure codes. This was originally proposed by Li et al. [13] but the early approach only considered the recovery performance not recovery bandwidth. The following ProMachine, ProHot, ProHot\_LazyCold are the novel methods proposed in this research which are the main contribution of this paper.
- **ProMachine:** When temporary long term machine failures are predicted with MTTR greater than 15 min, data in failure predicted machines are

proactively replicated to a dedicated node allocated specifically to handle temporary machine failure. In case of any failure, data is accessed from the dedicated node.

- **ProHot:** When temporary long term machine failures are predicted with MTTR greater than 15 min, data identified as hot in failure predicted machine will be proactively replicated to the dedicated node which has been allocated to handle temporary machine failure. In case of any failure, hot data is accessed from the dedicated node and typical reconstruction is applied to recover cold data.
- **ProHot\_LazyCold:** When temporary long term machine failures are predicted with MTTR greater than 15 min, data identified as hot in failure predicted machine is proactively replicated to a dedicated node that is allocated specifically to handle temporary machine failure. In case of any failure, hot data is accessed from the dedicated node and lazy recovery [12] is applied for cold data recovery.

This module is responsible for scaling up and down the number of dedicated temporary storage nodes, according to the failure predictions and amount of data need to be stored in temporary storage during a period of time. It is also responsible for allocating highly available node as a temporary storage such that any failure in this temporary storage node is minimal. Any failure prediction in this temporary storage will also lead to proactive replication. Any failure prediction in this temporary storage will also lead to proactive replication.

## 4.2 Recovery Approach

In our target scenario, a cloud storage system initially stores data with any  $(n,k)$  erasure code. With the help of disk/machine failure prediction methods employed in cloud storage systems, failure types and MTTR of node failures are predicted. Failures are also identified as disk, permanent machine, temporary long term machine (MTTR > 15 min), or temporary short term machine (MTTR < 15 min) failures. The set of data blocks  $(b_1, b_2, \dots, b_i)$  that is more likely to be accessed soon is defined as the hot data set  $H$ . Based on the failure types, hot data blocks, and client SLAs, one of the proposed recovery techniques ProDisk, ProMachine, ProHot, ProHot\_LazyCold will be chosen.

When the disk/permanent machine failures are predicted (proDisk), all the data blocks in the failure predicted disk (all data blocks of each disk in a failure predicted machine) are proactively replicated into the permanent storage as described in Procode [13]. The counter variables of corresponding replicated data blocks are incremented. These counter variables are used to identify if the particular data blocks are replicated already or to delete data blocks against noisy prediction. A delay is applied while deleting data blocks against noisy prediction. Time In Advance (TIA) which is provided by failure prediction algorithm is used as a time delay to delete the data blocks that are replicated due to noisy prediction. Time delay larger than TIA is the better choice. However,

this will result in extra storage. The choice of time delay varies and depends on the storage system where the system is utilized.

While temporary machine failures are predicted, proactive recovery is activated for either all (ProMachine) or some of the data blocks (ProHot, ProHot\_LazyCold) in a failure predicated machine. Data are replicated into the dedicated temporary storage. The data blocks that are not replicated are recovered by typical reconstruction of erasure codes. While data blocks are proactively replicated into temporary storage, the corresponding data blocks counter variables are incremented. These variables are used to identify if the particular data blocks are replicated already or to delete blocks when the machine recovers from temporary machine failures. The dynamic replication module also provisions and adjusts the number of temporary dedicated nodes, based on long term temporary machine failure rate and client SLAs. When the failure predicted nodes recover from actual failure and if no further failures are predicted for the same nodes, the proactively replicated data blocks corresponds to those nodes are deleted. Also, any data fragments which have more than one copy in the system are also deleted periodically. In the occurrence of node/disk failure, the reference is made to proactively replicated blocks which reduces number of data reconstructions in erasure coded storage systems.

## 5 Performance Analysis

Since all the methods proposed in this paper use a combination of proactive and lazy recovery methods, we will carry out the performance analysis on those methods.

### 5.1 Bandwidth Analysis

The bandwidth required to reconstruct any missing data is directly proportional to the number of transfers required, which is  $k$  in  $(n,k)$  erasure coded storage system. The amount of data transfer required to recover any missing block is

$$TransferRequired = S * (k + NumberOfMissingBlocks - 1) \quad (1)$$

where  $S$  is the chunk size and  $k$  is number of fragments needed to reconstruct data.  $k$  is 1 for replication. The recovery bandwidth is calculated as

$$RecoveryBandwidth = TransferRequired / RecoveryTime \quad (2)$$

Equation 2 shows that the RecoveryBandwidth is directly proportional to TransferRequired. Let us consider (14, 10) Reed-Solomon code with the chunk size of 250 MB. From Eq. 1, TransferRequired can be calculated as 2500 MB for recovering a single missing data block. However, it is 250 MB if the data block is proactively replicated. From this, we can conclude that proactive replication reduces the recovery bandwidth significantly. Lazy recovery delays the recovery of the data fragments until certain amount of data fragments are unavailable.



In this paper, we use lazy recovery only for handling long term temporary machine failures such that it does not impact durability of data. Since all the predicted disk failures are proactively replicated, it does not affect durability. Furthermore, lazy recovery is activated based on client SLA. If the client needs good read performance only for data identified as hot, it activates lazy recovery only for cold data. It also activates proactive recovery for hot data.

## 5.2 Storage Overhead Analysis

Erasure coding offers excellent storage efficiency compared to replication. Proportional increase in storage of various reliability methods is defined as:

$$(\text{systematicdata} + \text{originaldata})/\text{systematicdata} \quad (3)$$

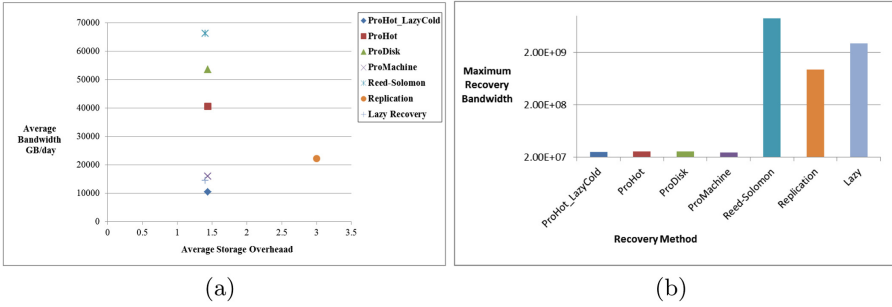
The method proposed in this paper proactively replicates data into a new hardware device when permanent node/disk failures are predicted. Once the failure predicted device fails, reference will be made to the proactively replicated device. Eventually, there will be wrong predictions about devices failing. When this occurs, it is expected that the storage overhead will suffer a slightly increase. False positive for disk failures are calculated as less than 0.1% using classification and regression trees [9]. Hence, the storage overhead will not be significantly increased by wrong predictions. Temporary nodes are dedicated to handle long term node failures. However, data in those temporary nodes are periodically evicted. Hence, temporary node failures will not increase storage overhead permanently.

## 6 Performance Evaluation

We use ds-sim simulator [12] to compare recovery bandwidth from replication and erasure coding to the various bandwidth efficient recovery technique proposed in this paper. We have simulated 3-tier storage components including disks, machines, and racks. We have modified ds-sim to add failure predictions, proactive replication, and hot data prediction. As output, ds-sim calculates repair bandwidth and number of degraded strips. The simulator models distributed storage systems of 3 Petabyte of storage for 10 years. Simulation parameters are 11 machines/rack, 20 disks/machine with each disk capacity of 750 GB and maximum recovery bandwidth capacity of 650 TB/day. Also 40% of random data blocks were considered as hot to evaluate ProHot and ProHot\_LazyCold recovery methods. For each result we run the simulation with number of iterations and calculated the result with 95% confidence interval.

### 6.1 Results and Discussions

In this section, we compare the bandwidth and reliability of replication, Reed-Solomon (14,10) and various recovery techniques proposed in this paper.



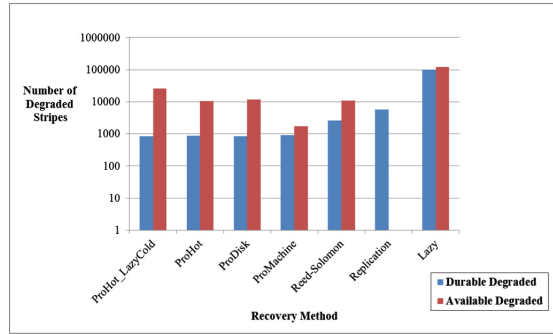
**Fig. 2.** (a) Average recovery bandwidth in GB per day and (b) Maximum instantaneous recovery bandwidth, in MB/hr, calculated over 10 years.

**Recovery Bandwidth.** We run simulations with the above configuration parameters with failure prediction rate 90%, false positive 0.1%, and time in advance 24 h which found reasonable in [9, 11]. Recovery bandwidth is calculated for each failure event except for machine failures lasting less than 15 min. Figure 2 shows the comparison of average recovery bandwidth in GB/day versus storage overhead for replication, Reed-Solomon(14,10), Lazy [12], and the various recovery techniques proposed in this paper. The proposed recovery techniques are also applied on Reed-Solomon (14,10) erasure code in this comparison.

Replication reduces recovery bandwidth in up to 66% compared to Reed-Solomon (14,10). ProDisk reduces average repair bandwidth up to 19% compared to Reed-Solomon (14,10). ProHot reduces recovery bandwidth up to 38% whereas ProMachine reduces recovery bandwidth by 75% compared to the same approach. ProMachine and ProHot\_LazyCold outperform replication. This is because in replication, data blocks are distributed among large number of hardware devices. Hence it experiences a large number of recovery events that increases recovery bandwidth. ProHot and ProHot\_LazyCold outperform lazy recovery. This is because the failure predicted hot data blocks are replicated proactively and it reduces number of lazy recoveries. However, ProMachine technique increases the temporary storage proportionally to the temporary long term machine failure rate.

Figure 2(b) shows the maximum instantaneous recovery bandwidth, in MB/hr (network traffic) in distributed storage systems over the simulation period. The simulation calculates network traffic as follows. Upon each recovery event, instantaneous total recovery bandwidth, in MB/hr is calculated and compared with the previous maximum recovery bandwidth. If the new recovery bandwidth is larger than maximum recovery bandwidth, the new recovery bandwidth becomes the maximum recovery bandwidth. The network traffic in (14,10) Reed-Solomon code is approximately 10 times higher than replication.

ProDisk, ProMachine, ProHot and ProHot\_LazyCold reduces network traffic better than replication and lazy recovery. This is due to proactive replication in erasure coding, which reduces amount of data to be transferred while keeping number of recoveries less than replication.



**Fig. 3.** Number of durable degraded and available degraded slices over 10 years.

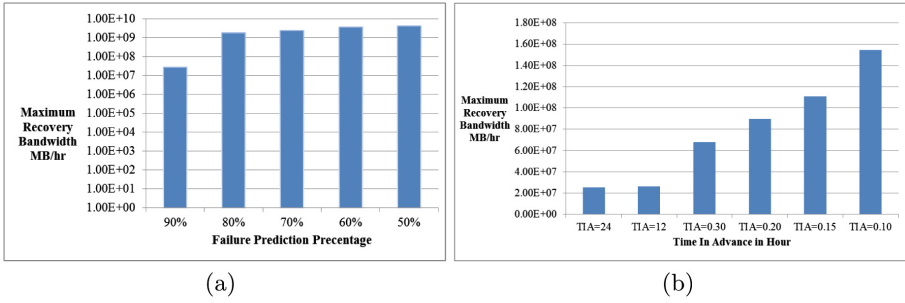
**Reliability.** To evaluate reliability of different approaches, we use the number of durable degraded slices and available degraded slices to compare durability and availability over the mission time. In a distributed storage systems, disks are partitioned into units called strip. Set of corresponding strips from  $n$  disks that encode and decode together is called stripe [18]. A stripe is termed degraded if one or more systematic blocks is unavailable. The term durable degraded refers the degraded stripe due to permanent failures, whereas available degraded refers to transient failures.

Replication does not increase available degraded slice counts in the system as request to any temporary unavailable slices are redirected to next available replica. Smaller number of durable and available degraded stripes indicates smaller probability of data loss as the system has less number of failure and repair events. Moreover, smaller number of degraded slices reduces the access latency and increases the performance of the application running on it. From Fig. 3 ProHot and ProHot\_LazyCold methods do not decrease number of available degraded stripes. However, available degraded slices are increased with respect to cold data. Also, the proposed system predicts and handles disk and node failures separately. ProHot and ProHot\_LazyCold methods handle all failure predicted disk failures proactively. Hence, they do not affect durability, contrary to lazy recovery method [12].

Proactively replicated data blocks reduce the number of durable degraded and available degraded slices in cloud storage systems and hence reduce the number of reconstructions. Less reconstructions reduces the number of data loss events in distributed storage. Figure 3 shows that even 90% of disk failure prediction rate do not eliminate degraded slices.

## 6.2 Sensitivity Analysis

The proposed recovery techniques are influenced by various important factors such as TIA and Failure Detection Rate. In this section, we examine how disk failure prediction rate affects network traffic and how the recovery bandwidth is affected by TIA.



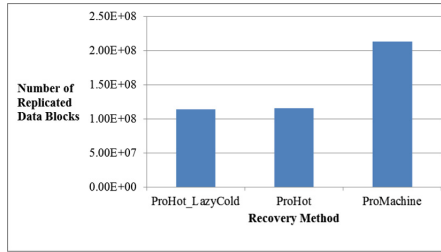
**Fig. 4.** Maximum instantaneous recovery bandwidth, in MB/hr, calculated over 10 years. (a) with varying failure prediction rates (b) for ProDisk with varying TIA.

**Disk Failure Prediction Rate.** For analyzing how the system is affected by the failure prediction rate, we measured network traffic with varying disk failure prediction rate. Li et al. [9], showed that more than 90% accuracy of disk failure prediction is possible. We run simulation with disk failure prediction accuracy varying from 50% to 90% and calculated recovery network traffic in ProDisk method, as shown in Fig. 4(a).

The proactive recovery in the storage systems will reduce network traffic (max instantaneous recovery bandwidth in MB/hr) associated with data reconstruction. As expected, network traffic decreases as the failure prediction rate increases. Accurate failure predictions proactively handle failures (transfer one data block instead of 10 data blocks in Reed-Solomon) in storage systems and hence reduce the recovery traffic. Moreover, only in the ProDisk the network traffic varies according to the prediction rate. The rest of the methods are accordance with machine failures. It transfers large amount of data while proactive recovery compared to ProDisk. Hence it is not showing much variations in network traffic with respective to prediction rates.

**Time in Advance.** We examine how the failure prediction's TIA affects recovery network traffic of storage systems. Figure 4(b) shows how the recovery network traffic changes with reduction of TIA of failure prediction in the ProDisk method. This will be similar for the rest of the methods. Since the maximum recovery bandwidth capacity in this experiments is set to 650 TB/day, reducing TIA from 24h to 12h does not change average recovery bandwidth drastically. However, reduction in TIA below 30 min increases network traffic in storage systems. Hence TIA will not affect the recovery bandwidth drastically.

**Amount of Data Transferred.** To evaluate resource savings from proactive replication only for hot data, we calculated the total amount of data transferred to the temporary dedicated storage to handle long term temporary machine failure. The amount of data transferred in ProHot/ProHot.LazyCold are directly proportional to the percentage of data determined as hot. Figure 5 shows that the



**Fig. 5.** Total number of proactively replicated slices due to long term temporary machine failures calculated over 10 years.

total amount of data transferred in ProMachine is approximately twice than in ProHot. The methods ProHot and ProHot\_LazyCold reduces temporary storage needs.

## 7 Conclusions and Future Work

The two primary reliability mechanisms employed by cloud storage systems have its own drawbacks. Even though erasure code offers tremendous storage savings compared to replication, reconstructing lost or corrupted data blocks involves large communication overhead.

In this paper, we proposed an approach that applies failure prediction techniques to proactively replicate and handle failures in erasure coded storage systems. We defined various recovery techniques with the combination of replication, erasure codes, and lazy recovery methods in order to reduce network bandwidth/traffic in cloud storage systems. It uses data blocks hot data status and client SLAs to define an appropriate recovery technique in cloud storage systems.

In our future work, we plan to investigate scheduling of proactive replicas in distributed storage such that it reduces degraded read latency in cloud storage. The interactions of foreground running tasks during proposed recovery schemes could also be considered in future. Another interesting and promising area of future research is energy-efficient scheduling of proactive replicas in cloud storage.

## References

1. Huang, C., et al.: Erasure coding in windows azure storage. In: Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC 12), Boston, USA, pp. 15–26. USENIX, June 2012
2. Nachiappan, R., Javadi, B., Calheiros, R.N., Matawie, K.M.: Cloud storage reliability for big data applications: a state of the art survey. *J. Netw. Comput. Appl.* **97**, 35–47 (2017)

3. Plank, J.S.: T1: erasure codes for storage applications. In: Proceedings of the 4th USENIX Conference on File and Storage Technologies (FAST), San Francisco, USA, pp. 1–74. USENIX, December 2005
4. Sathiamoorthy, M., et al.: XORing elephants: novel erasure codes for big data. *Proc. VLDB Endow.* **6**(5), 325–336 (2013)
5. Ford, D., et al.: Availability in globally distributed storage systems. In: Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), Vancouver, Canada. USENIX, October 2010
6. Dimakis, A.G., Godfrey, P.B., Wu, Y., Wainwright, M.J., Ramchandran, K.: Network coding for distributed storage systems. *IEEE Trans. Inf. Theory* **56**(9), 4539–4551 (2010)
7. Rashmi, K., Shah, N.B., Gu, D., Kuang, H., Borthakur, D., Ramchandran, K.: A hitchhiker’s guide to fast and efficient data reconstruction in erasure-coded data centers. *ACM SIGCOMM Comput. Commun. Rev.* **44**(4), 331–342 (2015)
8. Li, J., Stones, R.J., Wang, G., Liu, X., Li, Z., Xu, M.: Hard drive failure prediction using decision trees. *Reliab. Eng. Syst. Saf.* **164**, 55–65 (2017)
9. Li, J., et al.: Hard drive failure prediction using classification and regression trees. In: Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Atlanta, USA, pp. 383–394. IEEE, June 2014
10. Javadi, B., Kondo, D., Iosup, A., Epema, D.: The failure trace archive: enabling the comparison of failure measurements and models of distributed systems. *J. Parallel Distrib. Comput.* **73**(8), 1208–1223 (2013)
11. Agrawal, B., Wiktorski, T., Rong, C.: Analyzing and predicting failure in hadoop clusters using distributed hidden Markov model. In: Qiang, W., Zheng, X., Hsu, C.-H. (eds.) *CloudCom-Asia 2015*. LNCS, vol. 9106, pp. 232–246. Springer, Cham (2015). [https://doi.org/10.1007/978-3-319-28430-9\\_18](https://doi.org/10.1007/978-3-319-28430-9_18)
12. Silberstein, M., Ganesh, L., Wang, Y., Alvisi, L., Dahlin, M.: Lazy means smart: reducing repair bandwidth costs in erasure-coded distributed storage. In: Proceedings of the International Conference on Systems and Storage (SYSTOR), Haifa, Israel, pp. 1–7. ACM, June 2014
13. Li, P., Li, J., Stones, R.J., Wang, G., Li, Z., Liu, X.: Procode: a proactive erasure coding scheme for cloud storage systems. In: Proceedings of the 2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS), Budapest, Hungary, pp. 219–228. IEEE, September 2016
14. Li, W., Yang, Y., Yuan, D.: Ensuring cloud data reliability with minimum replication by proactive replica checking. *IEEE Trans. Comput.* **65**(5), 1494–1506 (2016)
15. Wu, S., Jiang, H., Mao, B.: Proactive data migration for improved storage availability in large-scale data centers. *IEEE Trans. Comput.* **64**(9), 2637–2651 (2015)
16. Wu, S., Jiang, H., Feng, D., Tian, L., Mao, B.: Improving availability of raid-structured storage systems by workload outsourcing. *IEEE Trans. Comput.* **60**(1), 64–79 (2011)
17. Liu, J., Shen, H.: A popularity-aware cost-effective replication scheme for high data durability in cloud storage. In: Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, pp. 384–389. IEEE, December 2016
18. Plank, J.S., Luo, J., Schuman, C.D., Xu, L., Wilcox-O’Hearn, Z., et al.: A performance evaluation and examination of open-source erasure coding libraries for storage. In: *FAST*, vol. 9, pp. 253–265 (2009)