Repairing Erasure Codes

Dimitris S. Papailiopoulos and Alexandros G. Dimakis
University of Southern California
Email: {papailio,dimakis}@usc.edu

***Abstract*—Distributed storage systems introduce redundancy to increase reliability. When erasure coding is used, the *exact repair problem* arises: if a node storing encoded information fails, in order to maintain the same level of reliability we need to create encoded information at a new node. This amounts to a partial recovery of the code, whereas conventional erasure coding focuses on the complete recovery of the information from a subset of encoded packets. The consideration of the repair network traffic gives rise to new design challenges. Recently, it was established that maintenance bandwidth can be reduced by orders of magnitude compared to standard erasure codes using network coding techniques. We discuss this work in progress and several theoretical and practical challenges on designing erasure codes specifically for storage systems when maintenance communication matters.**

I. Introduction

When deployed storage nodes are individually unreliable, redundancy must be introduced to improve availability and durability. Erasure coding techniques can potentially achieve orders of magnitude more reliability for the same redundancy compared to replication (see e.g. [3]). Cleversafe and Wuala are using erasure coding techniques and it seems that they are becoming more popular, especially for archival storage. Most storage systems use off-the-shelf erasure codes like Reed-Solomon codes and the related information dispersal algorithm of Rabin. Our research group focuses on coding theoretic designs and information theoretic bounds, specifically for distributed storage systems.

One specific problem we are currently focusing on is the challenge of maintaining an erasure encoded representation. We are specifically interested in maximum distance separable (MDS) erasure codes, with Reed–Solomon codes being perhaps the most popular example. Given two positive integers k and $n > k$, an (n, k) MDS code can be used for reliability: the data to be stored is separated into k information packets that are encoded into n packets (of the same size) such that *any* k out of these n suffice to recover the original data. An MDS encoded storage system can tolerate any $(n - k)$ node failures without data loss. It is important to note that each storage node can store multiple sub-packets that will be referred to as blocks (essentially using the idea of array codes [4], [5]).

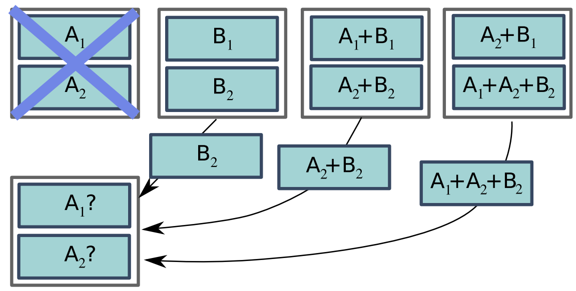

Fig. 1. Example of an (exact) repair: Assume that the first node in the previous storage system failed. The question is to repair the failure by creating a new node (the newcomer) that still forms a (4,2) MDS code. In this example it is possible to obtain exact repair by communicating 3 blocks, which is the information theoretic minimum cut-set bound.

The problem of interest is best illustrated through the example of Figure 1: Assume a file of total size $\mathcal{M} = 4$ blocks is stored using the $(4, 2)$ Evenodd code of the previous example and the first node fails. A newcomer node needs to construct and store two new blocks so that the three existing nodes combined with the newcomer still form a $(4, 2)$ MDS code. We call this the *repair problem* and focus on the required repair bandwidth. Clearly, repairing a single failure is easier than reconstructing all the data: since by assumption any two nodes contain enough information to recover all the data, the newcomer could download 4 blocks (from any two surviving nodes), reconstruct all four blocks and store A_1, A_2. However, as the example shows, it is possible to repair the failure by communicating only three blocks $B_2, A_2 + B_2, A_1 + A_2 + B_2$ which can be used to solve for A_1, A_2.

The repair problem and the corresponding regenerating codes were introduced in [10] and received some attention in the recent literature (see the online bibliography and the survey paper [1], [2]). The greatest interest streams from the fact that the fundamental lower bound on the total repair bandwidth outperforms the naive erasure code repair by roughly a $1/k$ factor. Surprisingly, there are new code constructions that can achieve this

$1/k$ factor reduction in network maintenance bandwidth, close or matching the aforementioned bound. This comes in sharp comparison with the straightforward application of Reed–Solomon or other existing codes *where the whole data object needed to be downloaded for the reconstruction of a single encoded block*. Erasure codes that match the information theoretic lower bounds for repair communication are known as regenerating codes.

II. The Status and Challenges of Regenerating Codes

In addition to the complete characterization of the information theoretic repair rate region there are several open problems we are working on: Most research on distributed storage has focused on designing MDS (or near-MDS) codes that are easily repairable. A different approach is to find ways to repair existing codes beyond the naive approach of reconstructing all the information. This is especially useful to leverage the benefits of known constructions such as reduced update complexity and efficient decoding under errors. The practical relevance of repairing a family of codes with a given structure depends on the applicability of this family in distributed storage problems. While the problem can be studied for any family of error correcting codes, two cases that are of special interest are Array codes and Reed-Solomon codes.

Array codes are widely used in data storage systems [5]–[7]. For the special case of Evenodd codes [4] a repair method that improves on the naive method of reconstructing the whole data object by a factor of 0.75 was established in [9]. There is still a gap from the cut-set lower bound and it remains open if the minimal repair communication can be achieved if we enforce the Evenodd code structure.

Another important family is Reed-Solomon codes. A repair strategy that improves on the naive method of reconstructing the whole data object for each single failure would be directly applicable to storage systems that use Reed Solomon codes. The repair of Reed-Solomon codes poses some challenges: since each encoded block corresponds to the evaluation of a polynomial, during repair a *partial evaluation* would have to be communicated from each surviving node.

Most of the prior work in this area has focused on the size of communicated packets. However, to create these packets, more information is actually read and merged by XORing before communication. In several cases, it may be that the amount of information that must be read from the storage nodes forms a bottleneck. It would be very interesting to characterize the minimum disk I/O required to repair an erasure code (see also [13]).

Finally, the issues of security and privacy are important for distributed storage. When coding is used, errors can be propagated in several mixed blocks through the repair process [11] and an error-control mechanism is required. A related issue is that of privacy of the data by information leakage to eavesdroppers during repairs [12].

References

[1] The Coding for Distributed Storage wiki `http://tinyurl.com/storagecoding`

[2] A. G. Dimakis, K. Ramchandran, Y. Wu, C. Suh, “A Survey on Network Codes for Distributed Storage” The Proceedings of the IEEE, (to appear).

[3] H. Weatherspoon and J. D. Kubiatowicz, “Erasure coding vs. replication: a quantitiative comparison,” in *Proc. IPTPS*, 2002.

[4] M. Blaum, J. Brady, J. Bruck, and J. Menon, “EVENODD: An efficient scheme for tolerating double disk failures in raid architectures,” in *IEEE Transactions on Computers*, 1995.

[5] M. Blaum, J. Bruck, and A. Vardy, “MDS array codes with independent parity symbols,” in *IEEE Transactions o Information Theory*, 1996.

[6] M. Blaum, P. G. Farrell, and H. van Tilborg, “Book chapter on array codes,” in *Handbook of Coding Theory, V. S. Pless and W. C. Huffman, Eds.*, 1998.

[7] C. Huang and L. Xu, “STAR: An efficient coding scheme for correcting triple storage node failures,” in *FAST-2005: 4th Usenix Conference on File and Storage Technologies*, (San Francisco, CA), December 2005.

[8] B. Marcus, R. M. Roth, and P. Siegel, “Constrained systems and coding for recording channels,” *Handbook of Coding Theory, V. Pless and W.C. Huffman (editors)*, pp. 1635–1764, 1998.

[9] Z. Wang, A. G. Dimakis, J. Bruck, “Rebuilding for Array Codes in Distributed Storage Systems,” Workshop on the Application of Communication Theory to Emerging Memory Technologies (ACTEMT) 2010.

[10] A. G. Dimakis, P. G. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, “Network coding for distributed storage systems,” IEEE Transactions on Information Theory, to appear.

[11] T. Dikaliotis and A. G. Dimakis and T. Ho “Security in Distributed Storage Systems by Communicating a Logarithmic Number of Bits”, *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, June 2010

[12] S. Pawar and S. El Rouayheb and K. Ramchandran, “On Security for Distributed Storage Systems,” *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, June 2010

[13] L. Xiang, Y. Xu, J. C. S. Lui and Q. Chang, “Optimal Recovery of Single Disk Failure in RDP Code Storage Systems,” ACM SIGMETRICS, June, 2010.