Non-homogeneous distributed storage systems

Vo Tam Van, Chau Yuen
Singapore Univ. of Tech. and Design
Email: {tamvan_vo,yuenchau@sutd.edu.sg}@sutd.edu.sg

Jing (Tiffany) Li
Lehigh University
Email: jingli@lehigh.edu

***Abstract*—This paper describes a *non-homogeneous* distributed storage systems (DSS), where there is one super node which has a larger storage size and higher reliability and availability than the other storage nodes. We propose three distributed storage schemes based on $(k+2,k)$ maximum distance separable (MDS) codes and non-MDS codes to show the efficiency of such non-homogeneous DSS in terms of repair efficiency and data availability. Our schemes achieve optimal bandwidth $\frac{k+1}{2}\frac{M}{k}$ when repairing 1-node failure, but require only one fourth of the minimum required file size and can operate with a smaller field size leading to significant complexity reduction than traditional homogeneous DSS. Moreover, with non-MDS codes, our scheme can achieve an even smaller repair bandwidth of $\frac{M}{2k}$. Finally, we show that our schemes can increase the data availability by 10% than the traditional homogeneous DSS scheme.**

***Index Terms*— Exact-repair MDS codes, non-homogeneous DSS, minimum storage regenerating (MSR) codes.**

I. Introduction

Distributed storage systems (DSS) are widely used today for storing data reliably over long periods of time using a distributed collection of storage nodes, which may be individually unreliable. Application scenarios include large data centers such as Total Recall [4], OceanStore [12] and peer-to-peer storage systems such as Wuala [9], that use nodes across the Internet for distributed file storage.

One of the challenges for DSS is the *repair problem*: If a node storing a coded piece fails or leaves the system, in order to maintain the same level of reliability, we need to create a new encoded piece and store it at a new node with the minimum repair bandwidth. To solve this problem, Dimakis *et al.* introduced a generic framework based on *regenerating codes* (RC) in [2]. RC use ideas from network coding to define a new family of erasure codes that can achieve different tradeoffs in the optimization of storage capacity and communication costs. The optimal tradeoff curve for achievable codes has two extremal points which are of particular interest: the minimum storage regenerating (MSR) codes with minimum possible storage size for a given repair capability, and the minimum bandwidth regenerating (MBR) codes with minimum possible repair bandwidth.

Consider a minimum storage system, where a source file of size M units is split into k parts, defined over a finite field $\mathbb{F}_q$ and stored across n nodes in the DSS. While the economy in storage is highly desirable, issues may arise when the system tries to repair failure at the optimal repair bandwidth. Specifically, if q or M grows to be arbitrarily large, then the system may become inefficient and impractical due to the high computational complexity or the fast growing storage consumption.

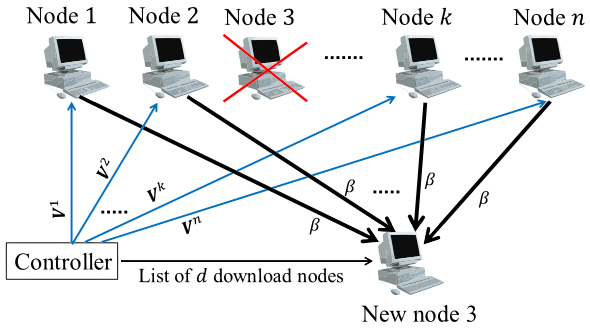

Figure 1. An example of traditional repairing 1-node failure.

Another challenge for DSS is data availability, which is of critical importance to a peer-to-peer (P2P) storage/backup system that relies on a swarm of distributed and independent nodes for file storage. As the nodes not only differ in their storage capacity and traffic bandwidth, but they may not be online or available at all times. Hence, there is a pressing need to increase the data availability, such that infomation is available with a probability approaching 1. Clearly, P2P enrironments are heterogeneous by nature, and code design for such systems must explicitly account for this heterogeneity.

The primary interest of this paper is to study a *non-homogeneous DSS*, where one *"super node"* has a larger storage size and higher reliability and availability than the other storage nodes. We study a class of high-rate $(k+2,k)$ MDS storage codes, and show that with MDS code such non-homogeneous DSS can achieve the optimal bound in [2] when repairing single or double-node failures, but require smaller M and q than the traditional homogeneous model in [1]. Another proposed scheme based on non-MDS codes is shown to repair 1-node failure below the optimal repair bandwidth bound in [2]. Moreover, we show that our proposed non-homogeneous DSS schemes can achieve a higher data availability than the traditional homogeneous DSS scheme.

This paper is organized as follows. Section II shows the definition of non-homogeneous DSS. Section III shows three proposed schemes of exact repair with $(k+2,k)$ storing codes in non-homogeneous DSS. Section IV shows the numerical results of our schemes and the comparison with previous methods. Finally, the paper is concluded in Section V.

II. Models of distributed storage systems

In this section, we present a brief review of the traditional homogeneous DSS proposed in [2]. Then, a new model of non-homogeneous DSS is proposed to realize the practical DSS.

A. Model of traditional homogeneous DSS

We follow the definition of traditional homogeneous DSS using $(n, k, d, \alpha, \gamma)$ regenerating codes over finite field $\mathbb{F}_q$. This network has n storage nodes and every k nodes suffice to reconstruct all the data. The size of the file to be stored is M units[1] and partitioned into k equal parts $\mathbf{f}_1, \cdots, \mathbf{f}_k \in \mathbb{F}_q^N$ where $N = \frac{M}{k}$. After encoding them into n coded parts using an (n, k) maximum distance separable (MDS) code, we store them at n nodes.

We define here the MDS property of a storage code using the notion of data collectors as presented in [2]. A storage code where each node contains $\frac{M}{k}$ worth of storage, has the MDS property if a data collector can reconstruct the all the M units by connecting to any k out of n storage nodes.

When a node fails, the data stored therein is recovered by downloading β packets each from any d $(\geq k)$ of the remaining $(n-1)$ nodes; the total repair bandwidth is then $\gamma = d\beta$ as shown in Fig. 1. It has been shown in [2] that there exists an optimal tradeoff between the storage per node, α, and the bandwidth to repair one node, γ. In this paper, we focus on the extreme point where the smallest $\alpha = \frac{M}{k}$ corresponds to a *minimum-storage regenerating* (*MSR*) code.

$$(\alpha_{MSR}, \gamma_{MSR}) = \left(\frac{M}{k}, \frac{Md}{k(d-k+1)}\right) \tag{1}$$

To minimize γ_{MSR}, let $d = n-1$ and we get $\left(\alpha_{MSR}, \gamma_{MSR}^{min}\right) = \left(\frac{M}{k}, \frac{M}{k}.\frac{n-1}{n-k}\right)$. In the case of high-rate codes $n = k+2$, a lower bound for repair bandwidth γ_1 of 1-node failure was shown as [2]:

$$\gamma_1 = (n-1)\beta \geq \frac{M}{k}.\frac{n-1}{n-k} = \frac{M}{k}.\frac{k+1}{2} \tag{2}$$

B. Model of the proposed non-homogeneous DSS

Definition 1. A non-homogeneous DSS with the parameter (n, k, h) is a distributed storage systems with h nodes based on (n, k) storing codes and the amount of data stored and downloaded from any nodes are variable. Node i in the network stores $\alpha_i \geq \frac{M}{k}$ units. When node i fails then it is repaired by downloading β_j packets from node j, $j \in \{n\} \setminus i$. □

It is clear that we must have $\beta_j \leq \alpha_j$ for all $j \neq i$ since a node can not transmit more information than it is storing. When $n = h, \alpha_i = \alpha, \beta_j = \beta$ for all i, $j \neq i$, we obtain the traditional homogeneous DSS. When $n > h$, there are more redundant blocks than the storage nodes. The storage process has to decide which node(s) to store more blocks.

Example 2. In this paper, we present the idea of non-homogeneous DSS using the following setting: there is one big node, called the *super node*, which has a larger storage capacity and higher reliability and availability than the other nodes. Such scenario is possible in practical system, e.g. in a peer-to-peer backup system, the super node could be the service provider that has higher availability and provides higher storage capacity than other peers.

Consider a system with one super node and three other storage nodes non-homogeneous DSS based on a $(5, 3)$ MDS code, which can be denoted as $(n = 5, k = 3, h = 4)$. Assume a file of size $M = 6$, then this file is divided into $k = 3$ parts, each part containing $N = \frac{M}{k} = 2$ packets. After encoding them into 5 encoded parts or 10 packets, we store the first $2N = 4$ packets in the super node, and each of the remaining three nodes stores $N = 2$ packets as shown in Fig. 2.

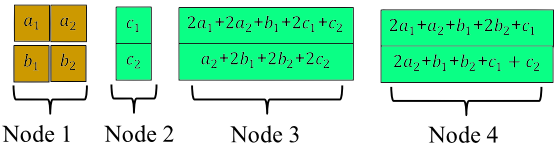

Figure 2. An example of non-homogeneous DSS based on $(5, 3)$ MDS codes and 4 storage nodes.(Node 1 is the super node.)

III. Exact repair of $(k+2, k)$ storing codes in non-homogeneous DSS

In this paper, we limit our study to high-rate $(n = k+2, k)$ exact-repair storing codes. This homogeneous problem has been considered in [1], [5], [14]. We propose three efficient DSS schemes using MDS and non-MDS storage codes in such $(n = k+2, k, h = k+1)$ non-homogeneous DSS, which are denoted as Scheme A, B, and C in Table. I. Scheme A and C use MDS codes while Scheme B uses non-MDS codes. The new system consists of $(k+1)$ nodes which include k nodes of storage size N and one super node of size $2N$.

A. Scheme A: Store two systematic data at the super node with MDS codes.

It can be seen in Table. I, the first $(k-1)$ storage nodes of Scheme A store the systematic file parts $\mathbf{f}_1, \cdots, \mathbf{f}_k$ where $\mathbf{f}_1, \mathbf{f}_2$ are stored in the same systematic node s_1 and the other file parts $\mathbf{f}_3, \cdots, \mathbf{f}_k$ are stored individually in the remaining $(k-2)$ systematic nodes $s_2, \cdots, s_{k-1}$, respectively. The first and second parity p_1, p_2 store a linear combination of all k systematic parts as $\mathbf{f}_1\mathbf{A}_1 + \cdots + \mathbf{f}_k\mathbf{A}_k$ and $\mathbf{f}_1\mathbf{B}_1 + \cdots + \mathbf{f}_k\mathbf{B}_k$. Here, $\mathbf{A}_i$ and $\mathbf{B}_i$ denote an $N \times N$ matrix of coding coefficients defined over finite field $\mathbb{F}_q$.

- **Repair systematic failure nodes for Scheme A**:

We first consider repairing 1-node failure (in the case of super node s_1 fails, it is considered as 2-node failures which will be discussed in more detail later). Without loss of generality we assume node s_2 that contains $\mathbf{f}_3$ is failed. For simplicity, we first consider the case $(n = 5, k = 3)$. To recover desired data $\mathbf{f}_3$, we have to download the following equations from the two survival parity nodes:

[1]We use "packets", "units", "blocks" interchangeably.

Table I

THREE SCHEMES OF $(k+2,k,k+1)$ NON-HOMOGENEOUS MODEL VS. TRADITIONAL MODEL BASED ON $(k+2,k)$ MDS CODES WHERE S AND P ARE THE ABBREVIATION OF SYSTEMATIC AND PARITY, RESPECTIVELY. HERE, $\mathbf{f}_i \in \mathbb{F}_q^{1\times N}$ AND $\mathbf{A}_i, \mathbf{B}_i \in \mathbb{F}_q^{N\times N}$ FOR ALL $1 \le i \le k$. NOTE THAT ALL SCHEMES A, B AND C USE ONLY $(k+1)$ STORAGE NODES TO STORE $(k+2)$ PACKETS.

	Non-homogeneous		Homogeneous
	Proposed Scheme A and B	Proposed Scheme C	Traditional model [1]
S. node			
s_1	$\mathbf{f}_1 \quad \mathbf{f}_2$	$\mathbf{f}_1$	$\mathbf{f}_1$
s_2	$\mathbf{f}_3$	$\mathbf{f}_2$	$\mathbf{f}_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_{k-1}	$\mathbf{f}_k$	$\mathbf{f}_{k-1}$	$\mathbf{f}_{k-1}$
s_k	$\times$	$\mathbf{f}_k$	$\mathbf{f}_k$
P. node			
p_1	$\mathbf{f}_1\mathbf{A}_1 + \cdots + \mathbf{f}_k\mathbf{A}_k$	$\mathbf{f}_1\mathbf{A}_1 + \cdots + \mathbf{f}_k\mathbf{A}_k$ $\mathbf{f}_1\mathbf{B}_1 + \cdots + \mathbf{f}_k\mathbf{B}_k$	$\mathbf{f}_1\mathbf{A}_1 + \cdots + \mathbf{f}_k\mathbf{A}_k$
p_2	$\mathbf{f}_1\mathbf{B}_1 + \cdots + \mathbf{f}_k\mathbf{B}_k$	$\times$	$\mathbf{f}_1\mathbf{B}_1 + \cdots + \mathbf{f}_k\mathbf{B}_k$

$$\begin{cases} \mathbf{f}_1\mathbf{A}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{A}_2\mathbf{V}^1 + \mathbf{f}_3\mathbf{A}_3\mathbf{V}^1 \\ \mathbf{f}_1\mathbf{B}_1\mathbf{V}^2 + \mathbf{f}_2\mathbf{B}_2\mathbf{V}^2 + \mathbf{f}_3\mathbf{B}_3\mathbf{V}^2 \end{cases} \tag{3}$$

where $\mathbf{A}_i, \mathbf{B}_i \in \mathbb{F}_q^{N\times N}$ for all $1 \le i \le k$ and $\mathbf{V}^1, \mathbf{V}^2 \in \mathbb{F}_q^{N\times \frac{N}{2}}$ can be derived based on the failure node. To repair different failure nodes, different $\mathbf{V}^1, \mathbf{V}^2$ are needed which can be precalculated. It can be seen from Fig. 3 that the term $\left(\mathbf{f}_1\mathbf{A}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{A}_2\mathbf{V}^1\right)$ and $\left(\mathbf{f}_1\mathbf{B}_1\mathbf{V}^2 + \mathbf{f}_2\mathbf{B}_2\mathbf{V}^2\right)$ are removable by downloading $\left(\frac{N}{2} + \frac{N}{2}\right)$ packets from super node. Therefore, the desired data $\mathbf{f}_3$ can be recovered if the following rank constraint is satisfied:

$$\text{rank}\left[\mathbf{A}_3\mathbf{V}^1,\ \mathbf{B}_3\mathbf{V}^2\right] = N \tag{4}$$

To recover the desired data $\mathbf{f}_3$ in the general $(n = k+2, k)$ case, we have to use the following equations:

$$\begin{cases} \mathbf{f}_1\mathbf{A}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{A}_2\mathbf{V}^1 + \mathbf{f}_3\mathbf{A}_3\mathbf{V}^1 + \cdots + \mathbf{f}_k\mathbf{A}_k\mathbf{V}^1 \\ \mathbf{f}_1\mathbf{B}_1\mathbf{V}^2 + \mathbf{f}_2\mathbf{B}_2\mathbf{V}^2 + \mathbf{f}_3\mathbf{B}_3\mathbf{V}^2 + \cdots + \mathbf{f}_k\mathbf{B}_k\mathbf{V}^2 \end{cases} \tag{5}$$

Similarly, the term $\left(\mathbf{f}_1\mathbf{A}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{A}_2\mathbf{V}^1\right)$ and $\left(\mathbf{f}_1\mathbf{B}_1\mathbf{V}^2 + \mathbf{f}_2\mathbf{B}_2\mathbf{V}^2\right)$ are removable by downloading $\left(\frac{N}{2} + \frac{N}{2}\right)$ packets from super node 1. The following conditions must be satisfied to achieve the optimal repair bandwidth:

$$\begin{aligned} \text{rank}\left[\mathbf{A}_3\mathbf{V}^1,\ \mathbf{B}_3\mathbf{V}^2\right] &= N \\ \text{rank}\left[\mathbf{A}_4\mathbf{V}^1,\ \mathbf{B}_4\mathbf{V}^2\right] &= \tfrac{N}{2} \\ &\vdots \\ \text{rank}\left[\mathbf{A}_k\mathbf{V}^1,\ \mathbf{B}_k\mathbf{V}^2\right] &= \tfrac{N}{2} \end{aligned} \tag{6}$$

To relax the complexity of the constraints found in (6), we set $\mathbf{A}_i = \mathbf{I}_N$ and $\mathbf{V}^1 = \mathbf{V}^2$, then obtain the following equations:

$$\begin{aligned} \text{rank}\left[\mathbf{B}_3\mathbf{V}^1,\ \mathbf{V}^1\right] &= N \\ \text{rank}\left[\mathbf{B}_4\mathbf{V}^1,\ \mathbf{V}^1\right] &= \tfrac{N}{2} \\ &\vdots \\ \text{rank}\left[\mathbf{B}_k\mathbf{V}^1,\ \mathbf{V}^1\right] &= \tfrac{N}{2} \end{aligned} \tag{7}$$

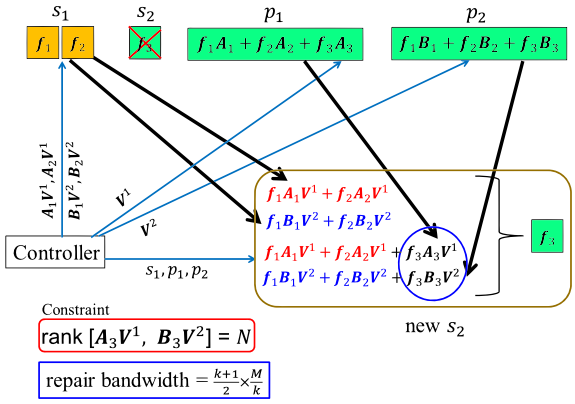

Figure 3. An example of repairing 1-node failure using Scheme A based on $(5,3)$ MDS codes in non-homogeneous DSS

The problem of finding matrix $\mathbf{B}_i$ is similar to [1]. However, here we only need to solve $(k-2)$ equations. Therefore, the fragment size and finite field will be smaller, with $M = 2^{k-1}k$ (which means the fragment size reduce to $\frac{1}{4}$ of the traditional homogeneous model when $k \ge 3$), and $q = 2k-1$. These advantages allow us to reduce the minimum size unit of storing file and reduce the complexity of computation to a smaller finite field. It can be seen that in this case of 1-node failure, the proposed Scheme A can achieve the optimal repair bandwidth of $\frac{k+1}{2}\frac{M}{k}$, which is the same as traditional homogenous DSS scheme.

- **Repair first parity node for Scheme A**:

If the first parity node p_1 fails, we make a change of variable to obtain a new representation for our code such that the first parity p_1 becomes a systematic node in the new representation. We make the change of variables as follows:

$$\sum_{i=1}^{k} \mathbf{f}_i = \mathbf{y}_3,\ \mathbf{f}_s = \mathbf{y}_s \quad \text{for } 1 \le s \ne 3 \le k \tag{8}$$

We solve (8) by replacing $\mathbf{f}_3$ in terms of the $\mathbf{y}_i$ variables and obtain

$$\mathbf{f}_3 = \mathbf{y}_3 - (\mathbf{y}_1 + \mathbf{y}_2 + \mathbf{y}_4 + \cdots + \mathbf{y}_k)$$

The problem of repairing first parity is equivalent to repair systematic node $\mathbf{y}_3$ in the new presentation. Note that $\mathbf{y}_1, \mathbf{y}_2$ are stored in the same node since they are correspondent to $\mathbf{f}_1, \mathbf{f}_2$. To repair $\mathbf{y}_3$, we have to download the following equations from node s_2 and p_2:

$$\begin{cases} (-\mathbf{y}_1) + (-\mathbf{y}_2) + \mathbf{y}_3 + \cdots + (-\mathbf{y}_k) \\ (\mathbf{B}_1 - \mathbf{B}_3)\mathbf{y}_1 + (\mathbf{B}_2 - \mathbf{B}_3)\mathbf{y}_2 + \mathbf{B}_3\mathbf{y}_3 + \cdots + (\mathbf{B}_k - \mathbf{B}_3)\mathbf{y}_k \end{cases}$$

Again, the $\mathbf{V}^1, \mathbf{V}^2$ matrices need to satisfy the following conditions in order to achieve the optimal repair bandwidth.

$$\begin{aligned} &\text{rank}\left[\mathbf{B}_3\mathbf{V}^1,\ \mathbf{V}^1\right] = N \\ &\text{rank}\left[(\mathbf{B}_4 - \mathbf{B}_3)\mathbf{V}^1,\ \mathbf{V}^1\right] = \tfrac{N}{2} \\ &\qquad\vdots \\ &\text{rank}\left[(\mathbf{B}_k - \mathbf{B}_3)\mathbf{V}^1,\ \mathbf{V}^1\right] = \tfrac{N}{2} \end{aligned} \tag{9}$$

Similar to the systematic case, the solution of matrix $\mathbf{B}_i$ is similar to [1]. However, we need to solve only $(k-2)$ equations, which means the fragment size and the finite field will be smaller $M = 2^{k-1}k$, and $q = 2k-1$.

- **Repair second parity node for Scheme A**:

Similar to the above, we rewrite this code in a form where the second parity is a systematic node in some presentation

$$\begin{bmatrix} \mathbf{I}_N & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{I}_N & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{I}_N & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{I}_N \\ \mathbf{I}_N & \mathbf{I}_N & \mathbf{I}_N & \cdots & \mathbf{I}_N \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 & \cdots & \mathbf{B}_k \end{bmatrix} \mathbf{f} = \begin{bmatrix} \mathbf{I}_N & 0 & 0 & \cdots & 0 \\ 0 & \mathbf{I}_N & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{I}_N & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \mathbf{I}_N \\ \mathbf{B}_1' & \mathbf{B}_2' & \mathbf{B}_3' & \cdots & \mathbf{B}_k' \\ \mathbf{I}_N & \mathbf{I}_N & \mathbf{I}_N & \cdots & \mathbf{I}_N \end{bmatrix} \mathbf{f}' \tag{10}$$

where $\mathbf{f}'$ is a full rank row transformation of $\mathbf{f}$. We proceed in a way similar to how we handled the first parity repair to achieve the optimal repair bandwidth. A similar set of equations to the case of repairing the first parity node can be obtained as shown below.

$$\begin{aligned} &\text{rank}\left[\mathbf{B}_3'\mathbf{V}^1,\ \mathbf{V}^1\right] = N \\ &\text{rank}\left[(\mathbf{B}_4' - \mathbf{B}_3')\mathbf{V}^1,\ \mathbf{V}^1\right] = \tfrac{N}{2} \\ &\qquad\vdots \\ &\text{rank}\left[(\mathbf{B}_k' - \mathbf{B}_3')\mathbf{V}^1,\ \mathbf{V}^1\right] = \tfrac{N}{2} \end{aligned}, \tag{11}$$

It can be seen that in this case the size and the finite field will be again $M = 2^{k-1}k$, and $q = 2k-1$, which are smaller than those in [1], and still achieve the optimal repair bandwidth.

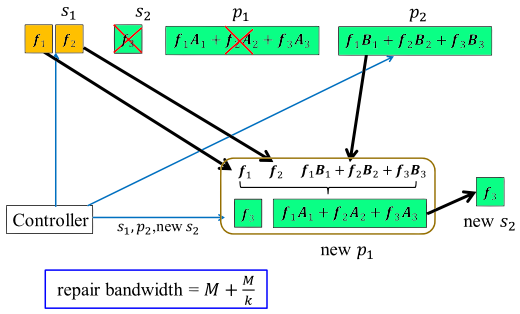

Figure 4. Illustration of exact repair of 2-node failure for a (5,3) Exact-Repair MDS code for Scheme A in non-homogeneous DSS. Total repair bandwidth $\gamma_2 = M + \frac{M}{k}$ achieves lower bandwidth bound.

- **Repair 2-node failures for Scheme A**

To repair 2-node failure at the optimal repair bandwidth, one solution is shown in Fig. 4. Lets assume that s_2 and p_1 fail, to repair them, first download the k packets from the survival nodes, then the original file can be recovered due to the property of MDS codes. Therefore, we can obtain the data of node s_2 and p_1, and store them in new node, say new p_1. Next, the data of the failure node s_2 (i.e. $\mathbf{f_3}$ in this case) is forwarded to the new node s_2. The total repair bandwidth will be $\gamma_2 = M + \frac{M}{k}$. It is trivial to repair the super node s_1 at the repair bandwidth of M by downloading data from survival nodes. It should be noted that the failure of one super node plus one additional node cannot be repaired since it can be regarded as three-node failure, therefore beyond the correcting ability of $(k+2, k)$ MDS codes.

B. Scheme B: Store two systematic data at the super node with non-MDS codes.

Scheme B uses the same model as Scheme A. However, we can achieve the repair bandwidth of 1-node failure below the optimal bound in this non-homogeneous model if the term $\left(\mathbf{f}_1\mathbf{A}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{A}_2\mathbf{V}^1\right)$ and $\left(\mathbf{f}_1\mathbf{B}_1\mathbf{V}^2 + \mathbf{f}_2\mathbf{B}_2\mathbf{V}^2\right)$ in (5) are the same or the following constraints are satisfied $\mathbf{A}_1\mathbf{V}^1 = \lambda\mathbf{B}_1\mathbf{V}^2, \mathbf{A}_2\mathbf{V}^1 = \lambda\mathbf{B}_2\mathbf{V}^2$. It means that we only need to download $\frac{N}{2}$ packets instead of $\left(\frac{N}{2} + \frac{N}{2}\right)$ packets from the super node to eliminate these terms. The following example is used to present the idea of repairing 1-node failure below the optimal bandwidth bound for the case $k = 3, n = 5$. Suppose $\mathbf{f}_1 = [a_1, a_2]^T, \mathbf{f}_2 = [b_1, b_2]^T, \mathbf{f}_3 = [c_1, c_2]^T$ and $\mathbf{p}_1 = \mathbf{f}_1\mathbf{A}_1 + \mathbf{f}_2\mathbf{A}_2 + \mathbf{f}_3\mathbf{A}_3$, $\mathbf{p}_2 = \mathbf{f}_1\mathbf{B}_1 + \mathbf{f}_2\mathbf{B}_2 + \mathbf{f}_3\mathbf{B}_3$ are the systematic and parity data of a $(5, 3)$ storage code over finite field $\mathbb{F}_3$ where

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} 2 & 0 \\ 2 & 1 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} 1 & 2 \\ 0 & 2 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}, \\ \mathbf{B}_1 &= \begin{bmatrix} 2 & 0 \\ 1 & 2 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}, \mathbf{B}_3 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{12}$$

It can be seen that any 1-node failure (systematic or parity node) except the super node can be repaired with bandwidth

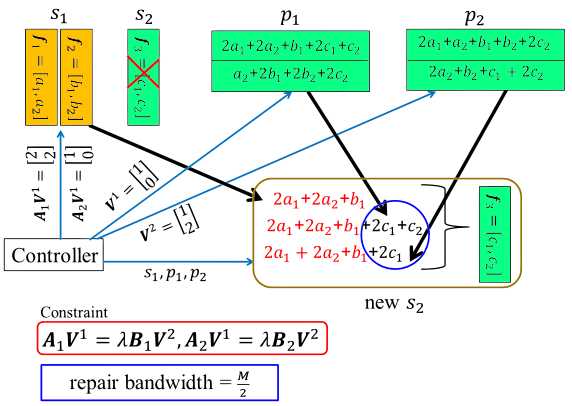

Figure 5. Total repair bandwidth of 1-node failure $\gamma_1 = \frac{M}{2}$ is smaller than the bound. In this example, $\gamma_1 = 3 < 4$ of repair bandwidth in the traditional case

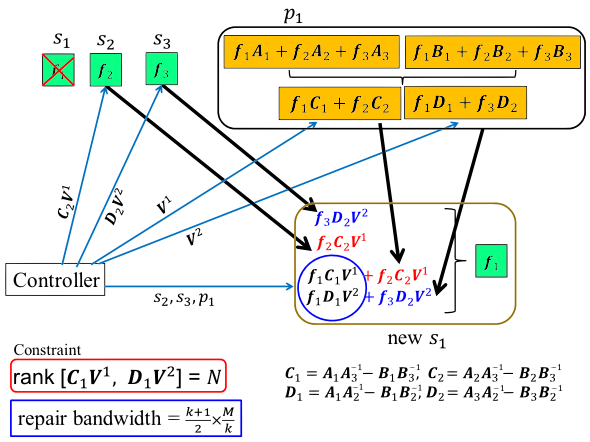

Figure 6. An example of repairing 1-node failure using Scheme C based on $(5, 3)$ MDS codes in non-homogeneous DSS

of $\frac{M}{2}$, which is below the optimal bound $\left(\frac{k+1}{2}\frac{M}{k}\right)$. Fig 5 shows the process of using two projection vectors $\mathbf{V}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{V}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ for repairing one systematic node failure below the optimal bandwidth bound. It is straightforward for the general case $(k+2, k)$. In the case of systematic 1-node failure, the general design constraints for Scheme B is:

$$\begin{aligned} \mathbf{A}_1\mathbf{V}^1 &= \lambda\mathbf{B}_1\mathbf{V}^2 \\ \mathbf{A}_2\mathbf{V}^1 &= \lambda\mathbf{B}_2\mathbf{V}^2 \\ \text{rank}\left[\mathbf{A}_3\mathbf{V}^1,\ \mathbf{B}_3\mathbf{V}^2\right] &= N \\ \text{rank}\left[\mathbf{A}_4\mathbf{V}^1,\ \mathbf{B}_4\mathbf{V}^2\right] &= \tfrac{N}{2} \\ &\vdots \\ \text{rank}\left[\mathbf{A}_k\mathbf{V}^1,\ \mathbf{B}_k\mathbf{V}^2\right] &= \tfrac{N}{2} \end{aligned} \tag{13}$$

Similar to Scheme A, we can find the solution for scheme B. However, in Scheme B the MDS property of the storage code breaks since we cannot reconstruct the original information from the survival nodes in the case of 2-node failure.

C. Scheme C: Store two parity data at the super node with MDS codes.

We first consider an example with $n = 5, k = 3$ for simplicity. Without loss of generality, assume that node 1 with data $\mathbf{f}_1$ is failed and the two parity packets $\mathbf{p}_1, \mathbf{p}_2$ are stored at the super node. To recover $\mathbf{f}_1$, we have the following equations after eliminating $\mathbf{f}_2$ and $\mathbf{f}_3$ from the parity node:

$$\begin{cases} \mathbf{f}_1\mathbf{A}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{A}_2\mathbf{V}^1 + \mathbf{f}_3\mathbf{A}_3\mathbf{V}^1 \\ \mathbf{f}_1\mathbf{B}_1\mathbf{V}^2 + \mathbf{f}_2\mathbf{B}_2\mathbf{V}^2 + \mathbf{f}_3\mathbf{B}_3\mathbf{V}^2 \end{cases} \rightarrow \begin{cases} \mathbf{f}_1\mathbf{C}_1\mathbf{V}^1 + \mathbf{f}_2\mathbf{C}_2\mathbf{V}^1 \\ \mathbf{f}_1\mathbf{D}_1\mathbf{V}^2 + \mathbf{f}_3\mathbf{D}_2\mathbf{V}^2 \end{cases} \tag{14}$$

where $\mathbf{C}_i, \mathbf{D}_i \in \mathbb{F}_q^{N\times N}$ for $i = 1, 2$ and $\mathbf{C}_1 = \mathbf{A}_1\mathbf{A}_3^{-1} - \mathbf{B}_1\mathbf{B}_3^{-1}, \mathbf{C}_2 = \mathbf{A}_2\mathbf{A}_3^{-1} - \mathbf{B}_2\mathbf{B}_3^{-1}, \mathbf{D}_1 = \mathbf{A}_1\mathbf{A}_2^{-1} - \mathbf{B}_1\mathbf{B}_2^{-1}, \mathbf{D}_2 = \mathbf{A}_3\mathbf{A}_2^{-1} - \mathbf{B}_3\mathbf{B}_2^{-1}$. It can be seen from Fig. 6 that the term $\mathbf{f}_2\mathbf{C}_2\mathbf{V}^1$ and $\mathbf{f}_3\mathbf{D}_2\mathbf{V}^2$ are removable by downloading $\left(\frac{N}{2} + \frac{N}{2}\right)$ packets from the parity node. Therefore, the desired data $\mathbf{f}_1$ can be recovered if the following rank constraint is satisfied:

$$\text{rank}\left[\mathbf{C}_1\mathbf{V}^1,\ \mathbf{D}_1\mathbf{V}^2\right] = N \tag{15}$$

For general $(k+2, k)$ case, similar to Scheme A, we set $\mathbf{A}_i = \mathbf{I}_N$ for all $i \le N$. To recover the desired data $\mathbf{f}_1$, we have the following equations reduction from parity node:

$$\begin{cases} \mathbf{f}_1\left(\mathbf{B}_1 - \mathbf{B}_3\right) + \mathbf{f}_2\left(\mathbf{B}_2 - \mathbf{B}_3\right) + \sum_{j=4}^{k} \mathbf{f}_j\left(\mathbf{B}_j - \mathbf{B}_3\right) \\ \mathbf{f}_1\left(\mathbf{B}_1 - \mathbf{B}_2\right) + \mathbf{f}_3\left(\mathbf{B}_3 - \mathbf{B}_2\right) + \sum_{j=4}^{k} \mathbf{f}_j\left(\mathbf{B}_j - \mathbf{B}_2\right) \end{cases} \tag{16}$$

The following conditions must be satisfied to achieve the optimal repair bandwidth of $\frac{k+1}{2}\frac{M}{k}$:

$$\begin{aligned} \text{rank}\left[\left(\mathbf{B}_1 - \mathbf{B}_2\right)\mathbf{V}^1,\ \left(\mathbf{B}_1 - \mathbf{B}_3\right)\mathbf{V}^2\right] &= N \\ \text{rank}\left[\left(\mathbf{B}_4 - \mathbf{B}_2\right)\mathbf{V}^1,\ \left(\mathbf{B}_4 - \mathbf{B}_3\right)\mathbf{V}^2\right] &= \tfrac{N}{2} \\ &\vdots \\ \text{rank}\left[\left(\mathbf{B}_k - \mathbf{B}_2\right)\mathbf{V}^1,\ \left(\mathbf{B}_k - \mathbf{B}_3\right)\mathbf{V}^2\right] &= \tfrac{N}{2} \end{aligned} \tag{17}$$

In general, solving (17) is still an open problem. Here, we give a numerical solution for the case $n = 6, k = 4$. Consider $\mathbf{f}_1 = [a_1, a_2]^T, \mathbf{f}_2 = [b_1, b_2]^T, \mathbf{f}_3 = [c_1, c_2]^T, \mathbf{f}_4 = [c_1, c_2]^T$ and $\mathbf{p}_1 = \mathbf{f}_1 + \mathbf{f}_2 + \mathbf{f}_3 + \mathbf{f}_4$, $\mathbf{p}_2 = \mathbf{f}_1\mathbf{B}_1 + \mathbf{f}_2\mathbf{B}_2 + \mathbf{f}_3\mathbf{B}_3 + \mathbf{f}_4\mathbf{B}_4$ are the systematic and parity data of a $(6, 4)$ storage code over finite field $\mathbb{F}_3$ where

$$\begin{aligned} \mathbf{B}_1 &= \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} 0 & 2 \\ 2 & 0 \end{bmatrix}, \\ \mathbf{B}_3 &= \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \mathbf{B}_4 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}. \end{aligned} \tag{18}$$

It can be seen that any 1-node failure except the super node can be repaired with optimal bandwidth $\left(\frac{k+1}{2}\frac{M}{k}\right)$. Fig. 7 shows the process of using two projection vectors $\mathbf{V}^1 = \mathbf{V}^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ for repairing the first systematic node.

For the case of super node p_1 and 2-node fail, the repair process is similar to scheme A. The total repair bandwidth will be M and $M + \frac{M}{k}$, respectively.

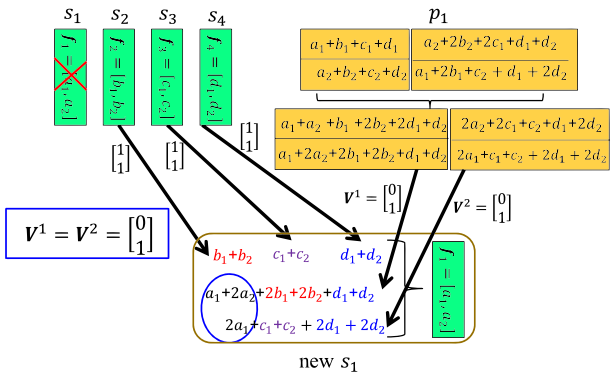

Figure 7. An example of repairing 1-node failure using Scheme C based on $(6, 4)$ MDS codes in non-homogeneous DSS

IV. Performance Analysis

As compared with the previous work in [1], our schemes A and C can achieve optimal repair bandwidth of 1-node failure at a smaller finite field q and 75% smaller data size M than [1]. Moreover, Scheme B that uses non-MDS code can repair 1-node failure with $\frac{M}{2k}$ smaller bandwidth than the optimal bound. A summary is presented in Table II for the various technologies.

A. Numerical Case Study

To show an illustration, we continue using the example $n = 5$, $k = 3$. Assume we have the data file of size 48 blocks to store across the DSS.

Scheme A and C: Divide the file into 4 fragments of size $M_1 = 12$ blocks. These fragments are stored across $k+1 = 4$ nodes in the non-homogeneous DSS. In the case of 1-node failure, the repair bandwidth will be $4 \times \frac{M_1}{k}\frac{k+1}{2} = 32$ blocks. In the case of 2-node failure, the repair bandwidth will be $4 \times \left(M_1 + \frac{M_1}{k}\right) = 64$ blocks. To update one fragment M_1 of the file, the update bandwidth will be $\frac{M_1}{k}n = 20$ blocks.

Scheme B: Similar to Scheme A, the file is divided into 4 fragments of size $M_1 = 12$. In the case of 1-node failure, the repair bandwidth will be $4 \times \frac{M_1}{2} = 24$. To update one fragment M_1 of the file, the update bandwidth will be $\frac{M_1}{k}n = 20$.

Alex method [1]: Divide the file into 1 fragment of size $M_2 = 48$. The repair bandwidth of 1-node failure will be $1 \times \left(\frac{M_2}{k}\frac{k+1}{2}\right) = 32$ and for 2-node failure requires $1 \times \left(M_2 + \frac{M_2}{k}\right) = 64$. The update bandwidth of one fragment M_2 will be $\frac{M_2}{k}n = 80$. The repair and update bandwidth of other methods are computed in the same manner and shown as in Table. III. Note that C.R.C method cannot repair 1-node failure with optimal bandwidth.

From Table. III, it can be seen that Schemes A and C can achieve the optimal bandwidth for repairing 1- or 2-node failure, which is the same as homogeneous DSS. By scarifying the MDS property, Scheme B requires a lower repair bandwidth for 1-node failure. It can be seen that all proposed schemes have an advantage of small update bandwidth in compare with the other schemes except the CRC method. However, the CRC method is not practical since it cannot achieve the optimal repair bandwidth in the case of 1-node failure. [5] and [14] methods are also impractical since they can repair only the systematic nodes.

B. Data Availability

In this subsection, we employ the framework proposed in [16] to measure and compare the data availability between our proposed non-homogenous DSS schemes and traditional homogenous DSS schemes to show the efficiency of our proposed schemes. Let $[p_1, \cdots, p_h]$ be the nodes' online probability of h nodes in the (n, k, h) DSS. Let the power set of h, 2^h, denote the set of all possible combinations of online nodes. Let $A \subset 2^h$ represents one of these possible combinations. Then, we will use Q_A to represent the event that combination A occurs. Since node availabilities are independent, we have

$$Pr[Q_A] = \prod_{i \in A} p_i \prod_{j \in 2^h \setminus A} (1 - p_j) \tag{19}$$

Let x_i be the number of data blocks stored in storage node i, for example $x_i = 1$, it means $\alpha_i = \frac{M}{k}$. The data allocation of our schemes will be $(x_1 = 2, x_2 = 1, \cdots, x_{k+1} = 1, x_{k+2} = 0)$. Let $L_k \subset 2^h$ be the subset containing those combinations of available nodes which together store k different redundant blocks.

$$L_k = \left\{ A : \ A \in 2^h, \sum_{i \in A} x_i \geq k \right\} \tag{20}$$

Since the retrieval process needs to download k different blocks out of the total n redundant blocks, the probability of successful recovery for an allocation $(x_1, \cdots, x_n)$ can be measured as

$$\begin{aligned} Pr\left[\text{successful recovery}\right] &= \textstyle\sum_{A \in L_k} Pr\left[Q_A\right] \\ &= \textstyle\sum_{A \in L_k} \left[\prod_{i \in A} p_i \prod_{j \in 2^h \setminus A} (1 - p_j)\right] \end{aligned} \tag{21}$$

To compare the data availability, we examine a scenario of node online probability where the online probability of super node is greater than the other node $p_1 \geq p_2 = p_3 = \cdots = p_n = p$. The data availability of homogeneous Pr_{homo} (e.g. scheme in [1]) and non-homogeneous $Pr_{non-homo}$ DSS (e.g. Schemes A and C, since Scheme B is based on non-MDS code, it is excluded in this study as its availability is calculated in a different manner) can be computed by the following equations:

$$Pr_{homo} = p^{k+1} + (k+1)(1-p)p^k + \frac{k(k+1)}{2} p_1 (1-p)^2 p^{k-1} \tag{22}$$

$$Pr_{non-homo} = p^k + kp_1(1-p)p^{k-1} + \frac{k(k+1)}{2} p_1 (1-p)^2 p^{k-1} \tag{23}$$

Let $p_1 = \chi p$ where $\chi \geq 1$. The condition $Pr_{non-homo} \geq Pr_{homo}$ will induce $\chi \geq p/(p + \frac{1}{2}(1-p)\left[(k-1) - (k+1)p\right])$. It can be seen that if $p \leq \frac{k-1}{k+1}$, then $p/(p + \frac{1}{2}(1-p)\left[(k-1) - (k+1)p\right]) \leq 1 \leq \chi$. Therefore, $Pr_{non-homo} \geq Pr_{homo}$ for all $p \leq \frac{k-1}{k+1}$. We run the simulations for the case

Table II

COMPARISON OF NON-HOMOGENEOUS MODEL VS. TRADITIONAL MODEL BASED ON $(k+2,k)$ CODES WHERE M AND γ ARE THE FILE SIZE AND REPAIR BANDWIDTH, RESPECTIVELY. SMALL VALUE OF M, γ AND q MEAN EFFICIENT.

	Scheme A&C	Scheme B	Alex [1]	Perm. code [5]	Tamo [14]	C.R.C [10]
	$M = 2^{k-1}k$ $q \geq 2k-1$	$M = 2^{k-1}k$ $q \geq 2k-1$	$M = 2^{k+1}k$ $q \geq 2k+3$	$M = 2^k k$ $q \geq 2k+1$	$M = 2^k k$ $q \geq 2k+1$	$M = 2k$ $q \geq n$
1- node failure	$\gamma = \frac{M}{k}\frac{k+1}{2}$	$\gamma = \frac{M}{2}$	$\gamma = \frac{M}{k}\frac{k+1}{2}$	$\gamma = \frac{M}{k}\frac{k+1}{2}$	$\gamma = \frac{M}{k}\frac{k+1}{2}$	N.A
2- node failures	$\gamma = M + \frac{M}{k}$	N.A	$\gamma = M + \frac{M}{k}$	$\gamma = M + \frac{M}{k}$	$\gamma = M + \frac{M}{k}$	$\gamma = M + \frac{M}{k}$

Table III

NUMERICAL RESULTS OF STORING A FILE OF SIZE 48 BLOCKS USING THE $(5,3)$ MDS CODES IN NON-HOMOGENEOUS AND HOMOGENEOUS DSS.

	Scheme A&C	Scheme B	Alex[1]	Perm. code[5]	Tamo [14]	C.R.C [10]
	$M_1 = 12$ $q \geq 5$	$M_1 = 12$ $q \geq 5$	$M_2 = 48$ $q \geq 9$	$M_3 = 24$ $q \geq 7$	$M_4 = 24$ $q \geq 7$	$M_5 = 6$ $q \geq 5$
1-node failure	$\gamma = 32$	$\gamma = 24$	$\gamma = 32$	$\gamma = 32$	$\gamma = 32$	N.A
2-node failures	$\gamma = 64$	N.A	$\gamma = 64$	$\gamma = 64$	$\gamma = 64$	$\gamma = 64$
update ≤ 12 data block	$\delta = 20$	$\delta = 20$	$\delta = 80$	$\delta = 40$	$\delta = 40$	$\delta = 10$

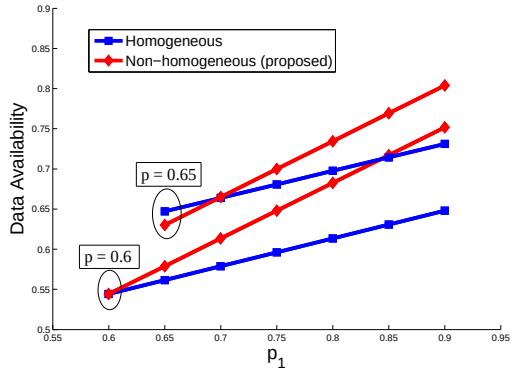

Figure 8. A comparision of data availability between non-homogeneous DSS and homogeneous DSS.

of $k = 4$, $p = 0.6$ and $p = 0.65$ and obtain the result in Fig. 8. It can be seen that for $p = \frac{k-1}{k+1} = 0.6$, data availability of non-homogeneous DSS scheme outperforms the homogeneous DSS scheme. For $p = 0.65 > \frac{k-1}{k+1}$, the non-homogeneous schemes also have a big improvement when p_1 has a high online availability. Therefore, it can be seen that our proposed non-homogeneous DSS schemes achieve a higher data availability than the traditional homogeneous DSS. The gap between the two becomes larger when the online availability of the super node increases, e.g. when p_1 is greater than 25% of p, the data availability of the proposed non-homogenous over homogenous DSS is increased by 10%.

V. CONCLUSIONS

We proposed three distributed storage schemes for *non-homogeneous DSS* with high rate $(k+2,k)$ codes. Two of the schemes make use of MDS code, and can achieve optimal repair bandwidth of $\frac{k+1}{2}\frac{M}{k}$ at smaller finite field q and 75% smaller fragment M than [2]. Small M and q are desirable, because they reduce the update bandwidth and complexity. Another scheme based on non-MDS code can achieve a smaller repair bandwidth than the optimal bandwidth based on MDS code by $\frac{M}{2k}$ for 1-node failure. We further demonstrate that in such non-homogeneous DSS, if we can ensure one super node with a higher online probability than the other nodes, we can achieve a higher data availability than the homogeneous DSS.

ACKNOWLEDGEMENT

This research is partly supported by the International Design Center (grant no. IDG31100102 & IDD11100101). Li's work is supported in part by the National Science Foundation under Grants No. CCF-0829888, CMMI-0928092, and EAGER-1133027.

REFERENCES

[1] D.S. Papailiopoulos, A.G. Dimakis, V.R. Cadambe, "Repair optimal erasure codes through Hadamard designs," *the 49th Annual Allerton Conference on Communication, Control and Computation*, pp.1382-1389, Sep. 2011.

[2] A.G. Dimakis, P. Godfrey, M. Wainwright, and K. Ramchandran, "Network coding for distributed storage systems," *IEEE Trans. Inform. Theory*, pp.4539-4551, Sep. 2010.

[3] Y.Wu, A.G. Dimakis, "Reducing repair traffic for erasure coding based storage via interference alignment," *IEEE International Symposium on Information Theory*, pp.2276-2280, July 2009.

[4] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G. M. Voelker, "Total recall: System support for automated availability management," *the 1st Symp. Networked Systems Design and Implementation (NSDI)*, Mar. 2004.

[5] V.R. Cadambe, C. Huang, J. Li, "Permutation code: optimal exact-repair of a single failed node in MDS code based distributed storage systems," *IEEE International Symposium on Information Theory*, pp.1225-1229, Aug. 2011.

[6] V.R. Cadambe, S.A. Jafar, H. Maleki, "Asymptotic interference alignment for exact repair in distributed storage systems," *the 44th Signals, Systems and Computers (ASILOMAR)*, pp.1617-1621, Nov. 2010.

[7] D. Cullina, A. G. Dimakis, and T. Ho, "Searching for minimum storage regenerating codes," *Allerton Conf. Control Comput. Commun.*, Sep. 2009.

[8] F. Dabek, J. Li, E. Sit, J. Robertson, M. Kaashoek, and R. Morris, "Designing a DHT for low latency and high throughput," *the 1st Symp. Networked Systems Design and Implementation (NSDI)*, Mar. 2004.

[9] D. Grolimund, "Wuala - A Distributed File System", Google Tech Talk http://www.youtube.com/watch?v=3xKZ4KGkQY8

[10] Cooperative regenerating codes, http://home.ie.cuhk.edu.hk/~wkshum/papers/CRC4.pdf
[11] N.B. Shah, K.V. Rashmi, P.V. Kumar, "A flexible class of generating codes for distributed storage," *IEEE International Symposium on Information Theory*, pp.1943-1947, Jun. 2010.
[12] S. Rhea, C. Wells, P. Eaton, D. Geels, B. Zhao, H. Weatherspoon, and J. Kubiatowicz, "Maintenance-free global data storage," *IEEE Internet Computer*, pp.40-49, Sep. 2011.
[13] C. Suh, K. Ramchandran, "Exact-repair MDS code construction using interference alignment," *IEEE Trans. Inform. Theory*, pp.1425-1442, Mar. 2011.
[14] I. Tamo, Z. Wang, J. Bruck, "MDS array codes with optimal rebuilding," *IEEE International Symposium on Information Theory*, pp.1240-1244, Aug. 2011.
[15] C.Armstrong, A. Vardy, "Distributed storage with communication cost," *the 49th Annual Allerton Conference on Communication, Control and Computation*, pp.1358-1365, Sep. 2011.
[16] L. Pamies-Juarez, P. Garcia-Lopez, M. Sanchez-Artigas, B. Herrera, "Towards the design of optimal data redundancy schemes for heterogeneous cloud storage infrastructures", Computer Network, pp. 1100-1113, Nov. 2011.