2009 10th International Symposium on Pervasive Systems, Algorithms, and Networks

# Reliability Analysis for Full-2 Code

Sheng Lin, Chi Zhang, Gang Wang, Xiaoguang Liu and Jing Liu
Nankai-Baidu Joint Lab, College of Information Technology Science, Nankai University
Weijin Road 94, Tianjin, China
Email: {shshsh.0510, arxor.kid}@gmail.com, wgzwp@163.com, liuxg74@yahoo.com.cn, jingliu@nankai.edu.cn

***Abstract*—Recently, with the fast development of storage system, 2-erasure coding schemes were widely used in industrial society. To meet different requirements, many kinds of 2-erasure coding schemes were presented, such as Reed-Solomon codes, binary linear codes, parity array codes, and so on. Full-2 code is a 2-erasure binary linear code. It is a non-MDS code, but achieves optimal encoding, decoding, and updating performance. Moreover, its fault tolerance is beyond 2, i.e. "2-erasure" is the huge undervaluation of its fault tolerance. It is hard to evaluate the precise reliability of full-2 code. The reason is that the reliability model is complex and the proportion of recoverable $k$-erasures ($k > 2$) to total $k$-erasures is difficult to calculate. In this paper, we present a combinatorial method to analyze the precise reliability of full-2 code. The reliability of full-2 based storage systems is also evaluated.**

## I. Introduction

In recent years, as hard disks have grown greatly in size and storage systems have grown in size and complexity, it is more frequent that a failure of one disk occurs in tandem with unrecovered failures of other disks or latent failures of blocks on other disks. On a system using single-erasure correcting code such as standard RAID-5, this combination of failures leads to a permanent data loss [1]. Hence, applications of multi-erasure correcting codes have become more pervasive. The multi-erasure codes are applicable to not only traditional disk arrays, but also data grids, peer-to-peer applications, digital fountains, and so on [2].

Researchers focused mainly on MDS (Maximum Distance Separable) codes in earlier studies. This kind of codes achieves optimal storage efficiency which implies minimum storage cost. By contrast, non-MDS codes generally have worse storage efficiency. However, this doesn't mean that non-MDS codes are useless. Non-MDS codes generally have better computational performance than MDS codes. Moreover, they are far more reliable than one might consider. Full-2 [3] is such a code. Although it is classified as a 2-erasure code, it can recover many 3-erasures, 4-erasures, and so on. Conventionally, Markov model is used to analyze MDS codes' reliability. But for non-MDS codes, Markov chains are extremely complex, and transition rates are difficult to calculate. This paper gives a combinatorial method to calculate transition rates. We analyze the reliability of full-2 RAID systems using this method.

## II. Related Work

An $m$-erasure code for a storage system is a scheme that encodes the content on $n$ data disks into $m$ check disks so that the system is resilient to any $m$ disk failures [2]. Unfortunately, there is no consensus on the best coding technique for general $n, m > 1$.

The known multi-erasure codes typically fall into one of the following three categories: *Reed-Solomon* (RS) codes, *binary linear codes* and *parity array codes*. RS codes [4] are the only known MDS codes for arbitrary $n$ and $m$. The computational complexity is a serious problem because RS codes are based on operations over Galois Field. Binary linear codes [3] are XOR-based, hence have perfect computational complexity. However, because they are non-MDS codes, bad storage efficiency is their inherent drawback. The key idea of this kind of codes is dividing data symbols (disks) into overlapped parity groups. Parity array codes can be regarded as a compact form of binary linear codes - they possess the XOR-based architecture, while packing data and parity symbols into fewer disks. Thus, parity array codes have good computational performance like linear codes, while achieving better storage efficiency. EVENODD [5], RDP [1], X-Code [6] and B-Code [7] are typical MDS array codes. Recently, non-MDS array codes such as Weaver codes [8] have attracted more attentions. They can be regarded as tradeoff between MDS array codes and binary linear codes.

Generally, the reliability of MDS codes is analyzed using Markov model [9]. The Markov chain in Fig. 1 describes the reliability model of 2-erasure MDS codes. The reliability of system and component (hard disks, network nodes, etc.) is assumed to obey exponential distribution. $\lambda = 1/MTTF$, where MTTF denotes the Mean Time To Failure of hard disks. $\mu = 1/MTTR$, where MTTR denotes the Mean Time To Repair of hard disks. The system is composed of $N$ disks. State 0 denotes the fault-free state. State 1 and 2 respectively denote the single-failure and 2-failure states. State $S$ denotes the data-loss state. The system MTTDL (Mean Time To Data Loss) can be calculated through this model.

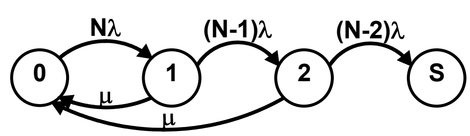


Fig. 1. Markov Chain for 2-erasure MDS Codes.

However, it is difficult to apply this method to non-MDS codes. First, $m$-erasure non-MDS codes generally can tolerate a part of $k$-erasures for some $k > m$. Thus Markov chains for $m$-erasure non-MDS codes contain more states than those for

978-0-7695-3908-9/09 $26.00 © 2009 IEEE
DOI 10.1109/I-SPAN.2009.95

$m$-erasure MDS codes. Second, the proportion of recoverable $k$-erasures ($k > m$) to total $k$-erasures is hard to calculate.

LDPC (*Low Density Parity Codes*) [10] is a special kind of binary linear codes. It has been widely used in communication field. Study on precise reliability of LDPC codes has a long history. Early study focused on asymptotical reliability of a family of LDPC codes instead of individual code instance. Recent works study concrete codes with optimal reliability [11], [12]. The methods are brute force search by computer or Monte Carlo simulation. The former is time consuming and the latter can not get precise results. Hafner analyzed the reliability of Weaver codes and LDPC codes in [13]. He also used brute force search. Elerath proposed an enhanced reliability mode for RAID systems using Monte Carlo simulation [14].

Our main contributions include: 1) studies the reliability of full-2 codes which is neglected by researchers before. Compared with LDPC codes, full-2 code has regular structure and better update penalty. We will show that it also has reasonable high reliability. 2) Presents a combinatorial method to analyze fault tolerance of full-2 codes precisely. 3) Evaluates the reliability of full-2 based RAID systems.

## III. Full-2 Code and Graph Representation

In a binary linear coding system, every bit in a disk participates in encoding, decoding and updating in the same way and independently. Therefore, for our purposes, "disk" (network node), "unit" (disk block) and "symbol" are equivalent. Full-2 is a code in which each data disk participates in exactly two parity groups, and each pair of parity groups share exactly one data disk [3]. Thus there are $\binom{n}{2}$ data disks in a full-2 coding system with $n$ parity disks, namely $\frac{n(n+1)}{2}$ disks in total. Fig. 2 shows the 10-disk full-2 code. $D_{ij}$ denotes the data disk that participates in the $i^{th}$ and the $j^{th}$ parity groups. $P_i$ denotes the parity disk of the $i^{th}$ parity group. The four parity groups are marked by four different hatches.

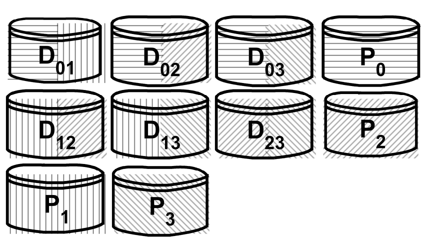


Fig. 2. 10-disk Full-2 Code.

We can represent a 2-erasure binary linear code by a simple graph: each node denotes a parity disk (group), each edge denotes a data disk, and the two vertices of an edge are just the two parity disks of the data disk [3], [7], [15]. Therefore the full-2 code with $n$ parity disks corresponds to $K_n$ - the complete graph with $n$ vertices. We have proven a theorem about reliability of binary linear codes [15].

*Theorem 1:* A binary linear coding system can recover from a $k$-erasure if and only if the $k$-erasure corresponds to a sub-graph that does not contain the following two types of structures:

1) A path and its two endpoints. We call this kind of unrecoverable erasure ***C**losed **P**arity **S**ymbols **S**ubset*, *CPSS* for short.
2) A cycle. We call it *CDSS* - ***C**losed **D**ata **S**ymbols **S**ubset*.

Note that concepts used here do not conform to conventional concepts used by graph theorists. In graph theory, an edge and its two endpoints are a one-piece unit. A sub-graph containing an edge means, of course, that it also contains the two vertices of the edge. However, if we represent a binary linear code by a simple graph, an edge being included in a sub-graph, does not mean that its two vertices are also included. The reason is that an edge denotes a data disk, and its two vertices denote the two parity disks the data disk belongs to. They are different entities. A data disk being involved in a $k$-erasure certainly does not mean that its two parity disks are also involved. So we modify the simple graph representation slightly. We introduce a new vertex, which is called the *parity sink vertex*, the *sink vertex* in short. We draw an edge between this vertex and every vertex else in the graph and let these new edges denote parity disks. So, vertices do not denote parity disks any longer, they only denote parity groups. Data and parity disks are both denoted by edges. Therefore, a $k$-erasure corresponds to a sub-graph composed of $k$ edges. The two kinds of unrecoverable structures mentioned in Theorem 1 are unified to be a cycle. We call this representation the *sink vertex representation*. Note that the full-2 code with $n$ parity disks is denoted by $K_{n+1}$ instead of $K_n$ using this representation. Fig. 3.a illustrates how to represent the 10-disk full-2 code by $K_5$.

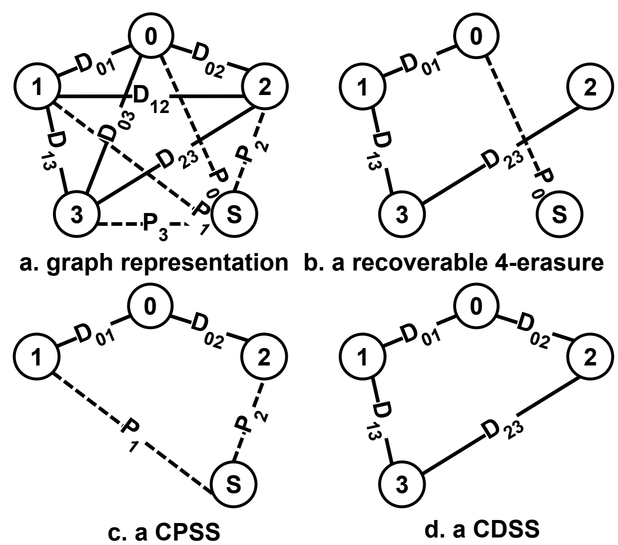


Fig. 3. Graph Representation.

Theorem 1 is intuitive. If a $k$-erasure is recoverable, there must be some data disk $D_{ij}$ that is the unique lost disk in one of its two parity groups, say, the $i$-th parity group. We can start decoding process from $D_{ij}$ by XORing all the other disks in group $i$. This makes group $j$ "single-loss". We then

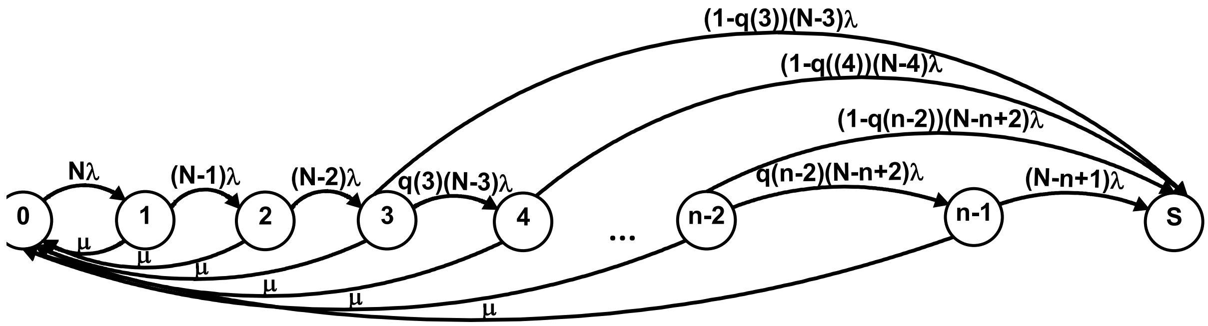


Fig. 4. Markov Chain for Full-2 Coding Systems.

reconstruct the unique lost disk, $D_{jk}$, in this group. Then the third failed disk is reconstructed in group $k$, and so on. This decoding chain corresponds to an open path in the sub-graph. Fig. 3.b gives an example. The sub-graph corresponds to a recoverable 4-erasure in the 10-disk full-2 code. Observe that $D_{23}$ should be reconstructed first because the second parity group only loses this data disk. So the third parity group has only one unknown member - $D_{13}$ - which is the second disk to be reconstructed. The next disk to be reconstructed is $D_{01}$. Finally $P_0$ is reconstructed. The decoding process goes along the open path $(2,3)$-$(3,1)$-$(1,0)$-$(0,S)$ in the sub-graph.

On the contrary, if a $k$-erasure is unrecoverable, there must be some disks can not be reconstructed. The only situation possible is that every involved parity group loses more than one disk. This implies that every edge in the corresponding sub-graph has more than one adjacent edge. Thus a cycle, i.e. a CPSS or a CDSS exists. Fig. 3.c and Fig. 3.d respectively show a CPSS and a CDSS in the 10-disk full-2 code. We can see that a CPSS corresponds to a cycle including the sink vertex, and a CDSS corresponds to a cycle excluding the sink vertex.

## IV. The Reliability Model

As mentioned in Section II, the reliability model of a 2-erasure MDS coding system appears as depicted in Fig. 1. Let $p_i(t)$ denote the probability system stays in state $i$ at time $t$, we have

$$
\begin{aligned}
\frac{\mathrm{d}p_0}{\mathrm{d}t} &= -N\lambda p_0(t) + \mu p_1(t) + \mu p_2(t) \\
\frac{\mathrm{d}p_1}{\mathrm{d}t} &= N\lambda p_0(t) - \mu p_1(t) - (N-1)\lambda p_1(t) \\
&= N\lambda p_0(t) - (\mu + (N-1)\lambda) p_1(t) \\
\frac{\mathrm{d}p_2}{\mathrm{d}t} &= (N-1)\lambda p_1(t) - \mu p_2(t) - (N-2)\lambda p_2(t) \\
&= (N-1)\lambda p_1(t) - (\mu + (N-2)\lambda) p_2(t)
\end{aligned}
\tag{1}
$$

The transition rate matrix is:

$$
M = \begin{pmatrix} -N\lambda & N\lambda & 0 \\ \mu & -\mu - (N-1)\lambda & N-1)\lambda \\ \mu & 0 & -\mu - (N-2)\lambda \end{pmatrix} \tag{2}
$$

According to Theorem 2 in [9], MTTDL of the system can be calculated by the following equation:

$$
MTTDL = \int_0^\infty p(t)\mathrm{d}t = -\vec{e} \cdot M^{-1} p(0) \tag{3}
$$

where $p(t)$ is the probability density function of system reliability, $\vec{e} = (1, 1, \cdots, 1)$, and $p(0) = (1, 0, \cdots, 0)$ is the initial state vector.

However, for non-MDS codes, things are more complicated. Since the fault tolerance of many $m$-erasure non-MDS codes is beyond $m$, the Markov chain contains more states than that of MDS codes. Moreover, for $k > m$, not 100% but a portion of $k$-erasures are recoverable. Thus state $k$ incidents to not only state $k+1$ and state 0, but also state $S$. The transition probabilities from state $k$ to state $k+1$ and from state $k$ to state $S$ share the probability of $(N-k)$ disks fail. Fig. 4 shows the Markov model for full-2 coding systems. $n$ denotes the number of parity disks. $N$ denotes the total number of disks. Thus $N = \binom{n+1}{2}$. We will show that the full-2 code with $n$ parity disks tolerate up to $n-1$ failures, thus the Markov chain contains $n+1$ states. $q(k)$ denotes the proportion of recoverable $k$-erasures to total $k$-erasures. This Markov chain is much harder to solve than that for 2-erasure MDS codes. Let $p_i(t)$ still denote the probability system stays in state $i$ at time $t$, we have:

$$
\begin{aligned}
\frac{\mathrm{d}p_0}{\mathrm{d}t} &= -N\lambda p_0(t) + \mu p_1(t) + \cdots + \mu p_{n-1}(t) \\
\frac{\mathrm{d}p_1}{\mathrm{d}t} &= N\lambda p_0(t) - \mu p_1(t) - (N-1)\lambda p_1(t) \\
&= N\lambda p_0(t) - (\mu + (N-1)\lambda) p_1(t) \\
\frac{\mathrm{d}p_2}{\mathrm{d}t} &= (N-1)\lambda p_1(t) - \mu p_2(t) - (N-2)\lambda p_2(t) \\
&= (N-1)\lambda p_1(t) - (\mu + (N-2)\lambda) p_2(t) \\
\frac{\mathrm{d}p_3}{\mathrm{d}t} &= (N-2)\lambda p_2(t) - \mu p_3(t) - (N-3)\lambda p_3(t) \\
&= (N-2)\lambda p_2(t) - (\mu + (N-3)\lambda) p_3(t)
\end{aligned}
\tag{4}
$$

For $k > 3$, we have:

$$\begin{aligned}\frac{\mathrm{d}p_k}{\mathrm{d}t} &= q(k-1)(N-k+1)\lambda p_{k-1}(t) - \mu p_k(t) \\ &\quad - (N-k)\lambda p_k(t) \\ &= q(k-1)(N-k+1)\lambda p_{k-1}(t) \\ &\quad - (\mu + (N-k)\lambda)p_k(t)\end{aligned} \tag{5}$$

The transition rate matrix is as Equation 6 shows. The analysis is base on the assumption that the fail disks can be recovered in parallel and recovery of multiple failures takes the same repairing time as that of single failure. We can see that what we need to do is to determine $q(k)$.

## V. Precise Fault Tolerance of Full-2 Code

Given a $k$, there are $\binom{N}{k}$ $k$-erasures in total in the $N$-disk full-2 code. What $k$-erasures are recoverable? According to Theorem 1, a $k$-erasure is recoverable if and only if its corresponding sub-graph contains no cycle. So we have the following lemma.

*Lemma 2:* A $k$-erasure is unrecoverable for all $k > n-1$.

*Proof:* Any sub-graph of $K_n$ with more than $n-1$ edges contains cycle. □

Now, we count the acyclic sub-graphs with $k$ edges, i.e. recoverable $k$-erasures, using a combinatorial method, for all $k > n$.

*Lemma 3:* In an indexed complete simple graph $K_n$, the number of $k$-edge acyclic sub-graphs equals to the number of spanning forest with exactly $n-k$ trees.

*Proof:* Given a $k$-edges acyclic sub-graph $X$ of $K_n$, we construct a spanning forest $Y$ by adding $K_n$'s vertex set of to it. Suppose that $Y$ contains $l$ trees. We connect the roots of the trees in $Y$ one-by-one. So a spanning tree is produced by adding $l-1$ edges to $Y$. Since a spanning tree of $K_n$ contains $n-1$ edges, we have $k+(l-1) = (n-1)$, thus $l = n-k$. Moreover, the mapping from $X$ to $Y$ is one-to-one. □

We denote by $g(n,k)$ the number of spanning forest of $K_n$ containing exactly $k$ trees. Now we calculate $g(n,k)$. We introduce some notations first.

We let $t(n)$ denote the number of spanning trees of $K_n$, $r(n)$ denote the number of rooted spanning trees, $f_k(n)$ denote the number of spanning forest containing $k$ trees, and $rf_k(n)$ denote the number of rooted spanning forest containing $k$ trees. The exponential generating functions of $r(n)$ and $t(n)$ are $R(x) = \sum_{n\geq 1} \frac{r(n)}{n!}x^n$ and $T(x) = \sum_{n\geq 1} \frac{t(n)}{n!}x^n$ respectively.

According to [16], we have the following three lemmas:

*Lemma 4:* $t(n) = n^{n-2}$, $r(n) = n^{n-1}$, $rf_k(n) = \binom{n-1}{k-1}n^{n-k}$.

*Lemma 5:* $R(x)\cdot e^{R(x)} = x$.

*Lemma 6:* The exponential generating functions of $f_k(n)$ and $rf_k(n)$ are $E_{f_k}(x) = \frac{[T(x)]^k}{k!}$ and $E_{rf_k}(x) = \frac{[R(x)]^k}{k!}$ respectively.

Based on these lemmas, we get the counting theorem.

*Theorem 7:* The number of spanning forest containing $k$ trees is $f_k(n) = n^{n-k}\sum_{i=0}^{k}(\frac{-1}{2n})^i\binom{n-1}{k+i-1}\binom{k}{i}\frac{(k+i)!}{k!}$.

*Proof:* Let

$$y = R(x) = \sum_{n\geq 1}\frac{r(n)}{n!}x^n = \sum_{n\geq 1}\frac{n^{n-1}}{n!}x^n$$

By Lemma 5 we have $y\cdot e^y = x$. Rewrite it as Lagrange expansion then we have

$$y - \frac{y^2}{2} = \sum_{n=1}^{\infty}\frac{n^{n-2}}{n!}x^n = T(x)$$

By Lemma 6, we have

$$E_{f_k}(x) = \frac{[T(x)]^k}{k!} = (y - \frac{y^2}{2})^k/k!$$

which can be rewritten as

$$E_{f_k}(x) = \frac{1}{k!}\sum_{i=0}^{k}(-\frac{1}{2})^i\binom{k}{i}y^{k+i}$$

According to Lemma 4 and 6, we have

$$y^r = \sum_{n=1}^{\infty}\frac{r!\binom{n-1}{r-1}n^{n-r}}{n!}x^n$$

So

$$\begin{aligned}[x^n]E_{f_k}(x) &= \frac{1}{k!}\sum_{i=0}^{k}(-\frac{1}{2})^i\binom{k}{i}([x^n]y^{k+i}) \\ &= \frac{1}{k!}\sum_{i=0}^{k}(-\frac{1}{2})^i\binom{k}{i}(\frac{(k+i)!\binom{n-1}{k+i-1}n^{n-k-i}}{n!}) \\ &= f_k(n)/n!\end{aligned}$$

which can be rewritten as

$$f_k(n) = n^{n-k}\sum_{i=0}^{k}(-\frac{1}{2n})^i\binom{n}{k+i-1}\binom{k}{i}\frac{(k+i)!}{k!}$$

□

Based on the above conclusions, we have some corollaries.

*Corollary 8:* For a full-2 system with $n$ parity disks, the proportion of recoverable $k$-erasures to total $k$-erasures is

$$\begin{aligned}\Pr(k) = \frac{1}{\binom{n(n+1)/2}{k}}(n+1)^k\sum_{i=0}^{n-k+1}(\frac{-1}{2(n+1)})^i \\ \binom{n}{n-k+i}\binom{n-k+1}{i}\frac{(n-k+i+1)!}{(n-k+1)!}\end{aligned}$$

*Proof:* By Lemma 3, $\Pr(k) = f_{n-k+1}(n+1)/\binom{N}{k}$, where $N = \binom{n+1}{2}$. Simply applying Theorem 7 we get the result. □

$$M = \begin{pmatrix} -N\lambda & N\lambda & & & & & & & \\ \mu & -\mu-(N-1)\lambda & (N-1)\lambda & & & & & & \\ \mu & & -\mu-(N-2)\lambda & (N-2)\lambda & & & & & \\ \mu & & & -\mu-(N-3)\lambda & q(3)(N-3)\lambda & & & & \\ \mu & & & & -\mu-(N-4)\lambda & & & & \\ \cdots & & & & & \cdots & & & \\ \mu & & & & & & -\mu-(N-n+3)\lambda & q(n-3)(N-n+3)\lambda & \\ \mu & & & & & & & -\mu-(N-n+2)\lambda & q(n-2)(N-n+2)\lambda \\ \mu & & & & & & & & -\mu-(N-n+1)\lambda \end{pmatrix} \quad (6)$$

We can also calculate the transition probability $q(k)$ from state $k$ to state $k+1$, i.e., when $k$ disks fail, the probability of one more disk failure doesn't leads to data loss).

*Corollary 9:* For $k \leq 3$,

$$q(k+1) = \frac{(k+1)f_{n-k}(n+1)}{(\binom{n+1}{2}-k)f_{n-k+1}(n+1)}$$

*Proof:* Let $A_k$ denote the event that the system can be recovered from a $k$-erasure. So

$$q(1) = q(2) = 1, q(3) = \Pr(3), q(k+1) = \mathrm{P}(A_{k+1}|A_k)$$

For a $k$-edge spanning forest of $K_{n+1}$, there are $\binom{n+1}{2} - k$ ways to add a new edge into the forest. Since there are $f_{n-k+1}(n+1)$ $k$-edge spanning forests in $K_{n+1}$, we need to consider $(\binom{n+1}{2}-k)f_{n-k+1}(n+1)$ cases in total. $f_{n-k}(n+1)$ of them remain spanning forests (with $k+1$ edges and $n-k$ trees). Moreover, each $(k+1)$-edge spanning forest is counted $k+1$ times. In conclusion,

$$q(k+1) = \frac{(k+1)f_{n-k}(n+1)}{(\binom{n+1}{2}-k)f_{n-k+1}(n+1)}$$

□

## VI. Reliability Analysis for Full-2 Systems

We analyzed the reliability of full-2 based storage systems using the method described in Section V. $\mu$ is set to $10^{-1}$, namely the repair time is about one day. Maple 11 is used to calculate Markov chains.

Fig. 5 shows MTTDL of full-2 based storage systems. We can see that if $\lambda$ is fixed, the reliability of full-2 systems decreases as system size climbs. But for tens of thousands of to millions of hours device MTTF (conform to the reliability of current enterprise hard disks), MTTDL of a full-2 system is longer than one million of hours even if the system contains more than 1000 devices. This reliability satisfies most applications.

Fig. 6 compares MTTDL of full-2 codes and MDS codes. All of the codes contain the same total number of disks. "MDS-2" denotes 2-erasure MDS codes. "MDS-n" denotes MDS codes containing the same number of parity disks as full-2 codes. "MDS=" denotes the MDS codes that have the similar MTTDL as full-2 codes. The right Y-axis is for this curve. It shows that, for a given total number of disks, how many parity disks a MDS code needs to achieve the almost

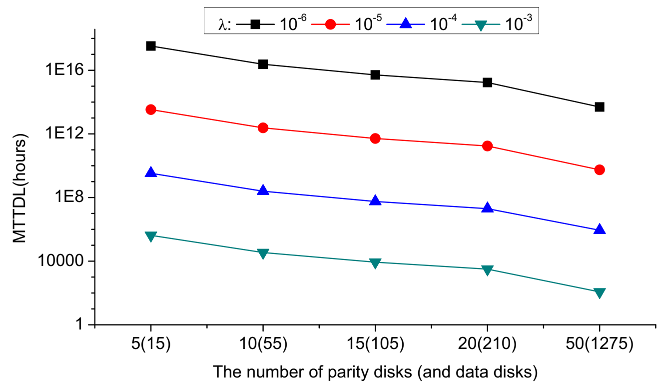


Fig. 5. MTTDL of full-2 based RAID.

same MTTDL as the full-2 code. We can see that although full-2 codes with $n$ parity disks are far inferior to $n$-erasure MDS codes in reliability, but they are far superior to 2-erasure MDS codes. Even for systems containing tens of thousands of devices, they provide reasonable reliability. Moreover, full-2 code is "MDS-increasing". Namely, as system size climbs, the curve "MDS=" ascends.

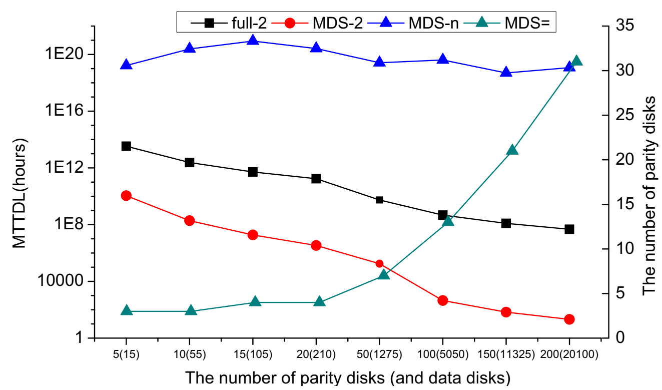


Fig. 6. full-2 vs. MDS codes.

Our evaluation shows that, for a big storage system, such as a distributed storage system with hundreds of nodes and thousands of hard disks, full-2 code is a good choice for reliability solution. Certainly, its MTTDL is lower than those of MDS codes and LDPC codes with the same number of parity disks. But in fact its reliability is good enough. MDS codes and LDPC codes oversupply reliability. On the other hand, full-2 code is superior to MDS codes and LDPC codes in performance. It's a 2-erasure code, thus its update penalty (the number of parity disks need to be update when a data disk is updated) is always 2 regardless of the number of parity disks $n$. While the update penalty of MDS codes is $n$. That of LDPC codes is very close to $n$. Moreover, the group size of full-2 codes with $N$ disks is $\Theta(\sqrt{N})$, while that of MDS codes and LDPC codes is $\Theta(N)$. Smaller group size means that fewer disks are involved during the recovery of a failed disk. That is to say, full-2 code is superior to MDS codes and LDPC codes in degrade- and reconstruct- mode performance. In addition, compared with "MDS=" codes, full-2 codes still have these advantages.

## VII. Conclusion and Future Work

Analyzing accurate reliability of non-MDS erasure codes is difficult. In this paper, we revised graph representation for full-2 codes. The new sink vertex representation translates $k$-erasure recoverability problem into sub-graph cyclic decision problem. Based on sink vertex representation, we presented a combinatorial method to calculate the proportion of recoverable $k$-erasures to total $k$-erasures precisely for all $k > 2$. We gave a precise analysis for full-2 based storage systems using this method. The analysis shows that full-2 code is a good reliability solution for large storage systems.

In the future, applying this methodology to other non-MDS erasure codes, such as $2d$-parity code, $3d$-parity code, full-3 code, Weaver code, and so on, is a valuable work. Analyzing reliability of other systems (such as peer-to-peer systems) based on non-MDS codes is also planned.

## Acknowledgements

This paper is supported partly by the National High Technology Research and Development Program of China (2008AA01Z401), NSFC of China (60903028), RFDP of China (20070055054), and Science and Technology Development Plan of Tianjin (08JCYBJC13000).

We would like to thank anonymous reviewers for their helpful comments.

## References

[1] P. F. Corbett, R. English, A. Goel, T. Grcanac, S. Kleiman, J. Leong, and S. Sankar, "Row-Diagonal Parity for Double Disk Failure Correction," in *Proceedings of the FAST '04 Conference on File and Storage Technologies*, San Francisco, California, USA, dec 2004, pp. 1–14.

[2] J. S. Plank, "Erasure Codes for Storage Applications," Tutorial Slides, presented at *the 4th Usenix Conference on File and Storage Technologies, FAST 2005*, San Francisco, CA, USA, December 2005.

[3] L. Hellerstein, G. A. Gibson, R. M. Karp, R. H. Katz, and D. A. Patterson, "Coding Techniques for Handling Failures in Large Disk Arrays," *Algorithmica*, vol. 12, no. 2/3, pp. 182–208, 1994.

[4] J. S. Plank, "A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-Like Systems," *Softw., Pract. Exper.*, vol. 27, no. 9, pp. 995–1012, 1997.

[5] M. Blaum, J. Brady, J. Bruck, and J. Menon, "EVENODD: An Efficient Scheme for Tolerating Double Disk Failures in RAID Architectures," *IEEE Trans. Computers*, vol. 44, no. 2, pp. 192–202, 1995.

[6] L. Xu and J. Bruck, "X-Code: MDS Array Codes with Optimal Encoding," *IEEE Transactions on Information Theory*, vol. 45, no. 1, pp. 272–276, 1999.

[7] L. Xu, V. Bohossian, J. Bruck, and D. G. Wagner, "Low-density MDS Codes and Factors of Complete Graphs," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 1817–1836, 1999.

[8] J. L. Hafner, "WEAVER Codes: Highly Fault Tolerant Erasure Codes for Storage Systems," in *Proceedings of the FAST '05 Conference on File and Storage Technologies*, San Francisco, California, December 2005, pp. 73–80.

[9] T. J. E. Schwarz, "Reliability and performance of disk arrays," Ph.D. dissertation, University of California at San Diego, La Jolla, CA, USA, 1994. [Online]. Available: http://portal.acm.org/citation.cfm?id=221861

[10] M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical Loss-Resilient Codes," in *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, May.

[11] J. S. Plank, A. L. Buchsbaum, R. L. Collins, and M. G. Thomason, "Small Parity-Check Erasure Codes - Exploration and Observations," in *2005 International Conference on Dependable Systems and Networks (DSN 2005)*, June.

[12] J. S. Plank and M. G. Thomason, "A Practical Analysis of Low-Density Parity-Check Erasure Codes for Wide-Area Storage Applications," in *2004 International Conference on Dependable Systems and Networks (DSN 2004)*, June.

[13] J. L. Hafner and K. Rao, "Notes on Reliability Models for Non-MDS Erasure Codes," IBM Research Division, Almaden Research Center, San Jose, CA, USA, Tech. Rep. RJ10391(A0610-35), Oct. 2006.

[14] J. G. Elerath and M. Pecht, "Enhanced Reliability Modeling of RAID Storage Systems," in *The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2007*, June.

[15] J. Zhou, G. Wang, X. Liu, and J. Liu, "The Study of Graph Decompositions and Placement of Parity and Data to Tolerate Two Failures in Disk Arrays: Conditions and Existance," *Chinese Journal of Computer*, vol. 26, no. 10, pp. 1379–1386, 2003.

[16] R. P. Stanley, *Enumerative Combinatorics Volume 2*. Cambridge, MA: Cambridge University Press, 1999.