Locating More Corruptions in a Replicated File

Ada Waichee Fu
Dept. of Computer Science and Engineering
The Chinese University of Hong Kong,
Shatin, Hong Kong
Email: adafu@cs.cuhk.hk

Siu-Cheung Chau
Dept. of Mathematics and Computer Science,
University of Lethbridge,
Lethbridge, Alberta, Canada, T1K 3M4
Email: chau@cs.uleth.ca

Abstract

When a data file is replicated at more than one site, we are interested in detecting corruption by comparing the multiple copies. In order to reduce the amount of messaging for large files, techniques based on page signatures and combined signatures have been explored. However, for 3 or more sites, the known methods assume that the number of corrupted page copies to be at most $\lceil M/2 \rceil - 1$, where M is the number of sites. We point out that this assumption is unrealistic and the corresponding methods are unnecessarily pessimistic. In this paper, we replace this assumption by another assumption which we show to be reasonable. Based on this assumption, we derived a distributed algorithm which in general achieves better performance than previously known results. Our system model is also more refined than previous work.

1. Introduction

When a data file is replicated at one or more sites in a distributed computer network, it is of interest to discover corruptions in the file copies by making comparison among the multiple physical copies of the file. Since a data file is typically large, it is inefficient to transmit the entire file to a remote site for comparison. The methods using *combined signatures* tackle this problem as follows. For each page copy, a *page signature* is computed. From the page signatures, *combined signatures* are computed and transmitted among sites. The corruptions are detected by performing some computations on a set of combined signatures. Each combined signature is derived from the entire set of page signatures, and typically the number of combined signatures needed for corruption detection is smaller than the total number of page signatures. Hence this method uses less signature transmission compared to methods using only page signatures.

Previous work considers the worst case scenario that the page signatures for corrupted copies can be identical. Due to this consideration, the number of corrupted page copies is assumed to be at most $\lceil M/2 \rceil - 1$, where M is the number of sites. We note that when the number of pages is large, which is a basic assumption motivating the method of combined signatures, this assumption is unrealistic. In this paper we argue that the chance of 2 page copies being corrupted and having identical page signatures is small and comparable to the tolerable probability of two differing pages having the same page signature. We thus assume that 2 corrupted page copies do not have the same signature. With the new assumption, we derive a distributed algorithm that needs to transmit a smaller number of signatures. A more refined Model is used, and it is shown that this can further enhance the performance of the algorithm.

The paper is organized as follows: Section 2 describes the problem and previous work. Section 3 is a description of some properties of the combined signatures that will be useful for the algorithm that follows. Section 4 presents the distributed algorithm together with the performance analysis and comparison with previous work. Section 5 is a conclusion.

2 Problem Definition

We consider the case where a file is replicated among a set of M sites $SS = \{s_1, ..., s_M\}$. We assume that the file contains N pages, $\{P_1, ..., P_N\}$, and the copy of page P_i at site s_j is $P_{i,j}$. Some previous work has assumed centralized control so that communication is always between a coordinator and the other sites. We assume decentralized control so that any site can communicate with any other site.

When there is no failure, the copies of each page are identical. When there is one or more site failures, some page copies may become corrupted. As in most previous papers, we assume that the total number of corrupted page copies will not exceed f. Furthermore, we assume that each site contains at most h corrupted page copies, and at most g sites has corruption. If these values are not known, then h is set to $\min\{N, f\}$, and g is set to the greatest possible value. The followings are notations used in this paper

N	number of pages in the file, $N < 2^b$
M	number of sites, $M \geq 3$
b	number of bits in each page signature
P_i	page i of the file
$P_{i,j}$	copy of page P_i at site s_j
$S(P_{i,j})$	page signature of page copy $P_{i,j}$
$sig_{j,m}$	a combined signature for site s_m where j is an integer
α	a primitive element in the finite field $GF(2^b)$ used to find a combined signature
f	maximum number of corrupted page copies in all sites at file comparison
g	maximum number of sites that contain corrupted page copies at file comparison $g \leq f$
h	maximum number of corrupted page copies at a site at file comparison $h \leq f$

Table 1 : Notations

The file size is assumed large, hence it is not cost-effective if the entire file copies are transmitted among the sites. A known approach is to to compute a *page signature* for each page copy, and transmit and compare the signatures instead. The page signature is typically much smaller than the page. For example [7] proposed to use a parity structure for the page signatures. Two pages with different page signatures are definitely not identical, but two pages that are different may happen to be translated into the same page signature. However the chance of this happening can be made small by making the signature size large. By exchanging the page signatures among the sites, it is possible to discover corruption.

However, as the number of page signatures may still be too large, some other methods would not send the page signatures directly. Instead, each site would compute a number of *combined signatures*. For example, the combined signature can be a bit-wise exclusive OR of a number of page signatures. By exchanging the combined signatures the corrupted pages are determined. Another way to compute the combined signatures is based on an error-correcting code over some finite field and is introduced in [8, 10]. In particular, a method using Reed-Solomon codes [5] has been suggested in [10] for problems with only 2 sites. [1] extended the method of Reed-Solomon codes and solves the problem for M replicated sites, where $M \geq 3$. Some other related work are [6, 4, 9, 3].

Although [1] solves a more general problem for M copies of the file, the following assumption is made.

Assumption Z: M $\geq 2f + 1$.

Assumption Z is made for the worst case consideration:

Consideration Z: *the page signatures for corrupted copies of a page can be identical.*

The condition in consideration Z implies one of the following cases:

1. two or more copies of the same page happen to be corrupted with identical error pattern, or
2. two or more copies of the same page are corrupted with different error patterns, but their page signatures turn out to be the same.

Note that the use of the method of page signatures comes with the tolerance of a problem related to the second case, in that a page copy may be corrupted so that its content is different from a correct copy, but the page signature generated for it happens to be identical to that of a correct copy, in which case, the method fails to identify the corruption.

If corrupted copies of a page can be identical, then it must be the case that $M \geq 2f+1$ in order to identify a correct version. This is because if $M < 2f + 1$, then it is possible that a majority number of sites fail at the same page, in which case a majority of the page signatures (the corrupted ones) for this page will be identical and we do not have a way to tell whether the corresponding copies are corrupted.

Let us look at some realistic figures for M and N. Suppose $M = 5, N = 2000$, with $M \geq 2f + 1$, f is at most 2, that is, Assumption Z says that at most 2 page copies can be corrupted. As there are in total 10,000 page copies, and memory corruption may occur in burst modes, one sees that this may not be a reasonable assumption. We conclude that Assumption Z is unrealistic, and that an algorithm based on this assumption and Consideration Z is probably a design with over-precaution. One would then consider if Consideration Z can be ignored. For example, can one instead assume that the page signatures for any 2 corrupted copies of the same page will not be identical? Here we show that such an assumption is reasonable. In particular, we shall support this assumption with the following arguments.

1. The probability $Prob_A$ that given any 2 corrupted page copies of the same page, their error patterns are identical is very small.
2. The probability $Prob_B$ that given any 2 corrupted page copies of the same page with different error patterns, their page signatures are the same is very small.
3. The use of page signatures has an inherent probability of assigning a corrupted page copy to the same page signature as a correct page copy, which has been claimed acceptable in previous work. Therefore, the probability $Prob_C$ that given a correct page copy and a corrupted page copy, they are given the same page signatures is assumed negligible. We shall show that $Prob_A$ and $Prob_B$ are no greater than $Prob_C$, and hence $Prob_A$ and $Prob_B$ should also be negligible.

Our argument is based on the assumption that site failures are independent. Suppose that each page of the file has b bits, then there are $2^b - 1$ different possible error patterns, we assume that the chance of occurrence of each pattern is the same.

Probability of a given corrupted page copy having a given error pattern (say pattern X) $= \frac{1}{2^b-1}$.

Probability of 2 given corrupted page copies having the same given error pattern $X = \frac{1}{(2^b-1)(2^b-1)}$.

Since there are $2^b - 1$ choices of the error patterns X in the above, probability of 2 given corrupted page copies having the same error pattern $= Prob_A = \frac{1}{(2^b-1)(2^b-1)}(2^b - 1) = \frac{1}{2^b-1}$. For reasonable values of b, this probability is very small.

Next we consider the value of $Prob_B$. Suppose that for an acceptable method, the probability that 2 different page copies being given identical page signatures is x. Therefore the probability $Prob_C$ that given a correct page copy and a corrupted page copy, they are given the same page signatures is x. The probability $Prob_B$ that given 2 corrupted page copies with different error patterns, their page signatures are the same is also x. Since $Prob_C$ is negligible, $Prob_B$ is also negligible.

Finally we point out that $Prob_A$ is comparable to acceptable values of $Prob_C$. For example, for the algorithm given in [7], suppose each signature has c bits, if the parity rules are chosen at random and the replicated file copies differ in a random fashion, there would be a probability of 2^{-c} that two page copies with different contents have the same page signature, or $Prob_C = \frac{1}{2^c}$ for this page signature method. The value of b is typically large compared to c, for example, in [7], the method is illustrated with $c = 40$ and $b = 2^{16} = 65536$. Therefore, $Prob_A = \frac{1}{(2^b-1)}$ is typically much smaller than $Prob_C$. Hence $Prob_A$ is negligible.

We would thus put forward the proposal that the probability of 2 pages being corrupted and having the same page signature is negligible. We shall see that this relaxation can lead to a new algorithm for locating corruption which has better performance. Our assumption is the following:

Assumption A: The probability that any 2 corrupted copies of the same page have identical page signatures is negligible. [1]

3. Properties of Combined Signatures

We shall adopt the combined signature techniques as in [10, 2, 1]. In this method, each site s_A calculates a page signature $S(P_{i,A})$ for each of its page copies $P_{i,A}$. For this method to work, it must be the case that $N < 2^b$, where b is the number of bits of a page signature. Since there are 2^b possible values of page signatures, each page signature can be considered as an element in a finite field (Galois field) $GF(2^b)$. From these page signatures, s_A can further compute **combined signatures** of b bits.

[1] Assumption A is based on a more basic assumption: *site failures are independent.* It may not hold if, for example, some copies are obtained from duplicating another. However, in such cases, it is not obvious that a feasible corruption detection method exists, since a majority of the copies may be corrupted.

Suppose a site s_A with a file copy $F_A = < P_{1,A}, ..., P_{N,A} >$ computes k combined signatures. These combined signatures are given by

$$sig_{j,A} = \sum_{i=1}^{N} \alpha^{ij} S(P_{i,A}) \quad \text{for } j = 0, 1, ..., k-1,$$

where α is a primitive element in $GF(2^b)$ and the addition and multiplication operations are those of $GF(2^b)$. Hence the elements α^j, for $1 \leq j \leq 2^b - 1$, are distinct. Note that $\alpha^{2^b-1} = \alpha^0 = 1$ and $\alpha^{2^b} = \alpha$. There are 2^b possible values for a combined signature. We refer to the signatures $sig_{j,A}$ for $j = 0, 1, 2, ...k-1$, where $k \geq 1$, as the **first** k combined signatures.

To help explain the algorithm in the next section, we first describe a few properties of the combined signatures. In the algorithm we shall compare two sets of combined signatures from two different sites s_A and s_B. In a special case, suppose we compute the first N combined signatures in each site, and we know that one of the sites, say s_A, is correct. The two sets of signatures are given by

$$sig_{j,A} = \sum_{i=1}^{N} \alpha^{ij} S(P_{i,A}) \quad \text{for } j = 0, 1, ..., N-1$$
$$sig_{j,B} = \sum_{i=1}^{N} \alpha^{ij} S(P_{i,B}) \quad \text{for } j = 0, 1, ..., N-1$$

If site s_A sends its first N combined signatures to site s_B, then s_B can compute the difference between $sig_{j,A}$ and $sig_{j,B}$ for $j = 0, 1, ..., N-1$. The difference between the two sets is given by

$$sig_{j,A} - sig_{j,B} \;=\; \sum_{i=1}^{N} \alpha^{ij} (S(P_{i,A}) - S(P_{i,B})) \quad (1)$$

for $j = 0, 1, ..., N-1$. Equation (1) can be written in a matrix form:

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \alpha^2 & \alpha^4 & \cdots & \alpha^{2N} \\ \vdots & \vdots & & \vdots \\ \alpha^{(N-1)} & \alpha^{2(N-1)} & \cdots & \alpha^{N(N-1)} \end{pmatrix} \begin{pmatrix} S(P_{1,A}) - S(P_{1,B}) \\ S(P_{2,A}) - S(P_{2,B}) \\ \vdots \\ S(P_{N,A}) - S(P_{N,B}) \end{pmatrix} = \begin{pmatrix} sig_{0,A} - sig_{0,B} \\ sig_{1,A} - sig_{1,B} \\ \vdots \\ sig_{N-1,A} - sig_{N-1,B} \end{pmatrix}$$

Let $e_i = S(P_{i,A}) - S(P_{i,B})$. Then e_i is 0 if and only if the signatures $S(P_{i,A})$ and $S(P_{i,B})$ are identical. Let $sig_{j,A} - sig_{j,B} = E_j$. We see that E_j is in $GF(2^b)$. Then we have

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \alpha^2 & \alpha^4 & \cdots & \alpha^{2N} \\ \vdots & \vdots & & \vdots \\ \alpha^{(N-1)} & \alpha^{2(N-1)} & \cdots & \alpha^{N(N-1)} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_N \end{pmatrix} = \begin{pmatrix} E_0 \\ E_1 \\ E_2 \\ \vdots \\ E_{N-1} \end{pmatrix}$$

In the above, the left most matrix is a **Vandermonde matrix**. A Vandermonde matrix is a matrix of the form

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ X_1 & X_2 & \cdots & X_k \\ X_1^2 & X_2^2 & \cdots & X_k^2 \\ \vdots & \vdots & & \vdots \\ X_1^{k-1} & X_2^{k-1} & \cdots & X_k^{k-1} \end{pmatrix}$$

Theorem 7.2.1 of [5]: The above Vandermonde matrix has a nonzero determinant if and only if all of the X_i for $i = 1, ..., k$ are distinct. □

Since the elements α^j are distinct, for $1 \leq j < 2^b$, and we assume $N < 2^b$, therefore the left most matrix in the above equation is nonsingular, and hence given E_j's, there is a unique solution for e_i's. If site s_A contains no corruption, then the non-zero e_i's identify the locations of page corruptions for site s_B.

This property has been stated as follows:

Remark 1 [1]: For any given $E_0, ..., E_{N-1} \in GF(2^b)$, the system of equations $\sum_{n=1}^{N} e_n \alpha^{jn} = E_j$, where $j = 0, ..., N-1$, has a unique solution for $(e_1, ..., e_N)$. □

Observation 1: Given the first N combined signatures $sig_{i,C}$ of a correct sites s_C, for $i = 0, ...N-1$, and the first N combined signatures $sig_{i,A}$ of a site s_A, one can solve the set of equations:

$$\sum_{n=1}^{N} e_n \alpha^{nj} = sig_{j,C} - sig_{j,A}, \quad for j = 0, ...N-1 \qquad (2)$$

to identify the location of all corruptions in s_A. □

Observation 1 tells us how to detect errors in a special case given a correct site. However, in the beginning we do not know any correct site, a correct site must first be discovered. Here we give a property of the combined signatures that allows us to do so without the need of sending too many combined signatures. We shall show that if we have at most v errors located at a set of 2 sites, we need only compare the first v combined signatures of two sites to determine if they are both correct or if at least one contains error. This property has been noted as follows:

Remark 2 [1]: Suppose that $(e_1, ..., e_N)$ has at most v non-zero entries. If $\sum_{n=1}^{N} e_n \alpha^{jn} = 0$, where $j = 0, ..., v-1$, then $e_n = 0$ for $n = 1, ..., N$. □

The set of equations in Remark 2 can be written as

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \vdots & \vdots & & \vdots \\ \alpha^{v-1} & \alpha^{2(v-1)} & \cdots & \alpha^{N(v-1)} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (3)$$

To see Remark 2, we note that the above equation can be simplified if at most v of the e_i's are non-zero. We simplify by retaining only the columns C_i in the left most matrix for $e_i \neq 0$. Suppose the columns indexed $i_1, i_2, ..i_t$ where $t \leq v$ remains. We form a square matrix by further extracting only the rows with indices $i_1, i_2, ..i_t$ as in the following set of equations.

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{(i_1)i_1} & \alpha^{(i_2)i_1} & \cdots & \alpha^{(i_t)i_1} \\ \vdots & \vdots & & \vdots \\ \alpha^{(i_1)(i_{t-1})} & \alpha^{(i_2)(i_{t-1})} & \cdots & \alpha^{(i_t)(i_{t-1})} \end{pmatrix} \begin{pmatrix} e_{i_1} \\ e_{i_2} \\ \vdots \\ e_{i_t} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

We need only observe that $e_{i_j} = 0$ for $0 \leq j \leq t-1$ is a trivial solution to the above set of equations, which are also part of the solutions for the set of equation in Remark 2. The above matrix is a sub-matrix of the previous Vandermonde matrix, one can show that this sub-matrix is non-singular by a similar proof as that for Theorem 7.2.1 in [5]. Hence there is exactly one solution for the above matrix.

The left hand side of equations (3) can be written as

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \vdots & \vdots & & \vdots \\ \alpha^{v-1} & \alpha^{2(v-1)} & \cdots & \alpha^{N(v-1)} \end{pmatrix} \begin{pmatrix} S(P_{1,A}) - S(P_{1,B}) \\ S(P_{2,A}) - S(P_{2,B}) \\ \vdots \\ S(P_{N,A}) - S(P_{N,B}) \end{pmatrix}$$

Therefore, if we compare the first v combined signatures for s_A and s_B and find that $sig_{j,A} = sig_{j,B}$ for $1 \leq j \leq v$, then the signatures $S(P_{i,A})$ and $S(P_{i,B})$ are identical for all i's, and therefore the page copies $P_{i,A}$ and $P_{i,B}$ are also identical. Since we assume (Assumption B) that two page copies of the same page that are corrupted are not identical, it can only be the case that sites s_A and s_B are both correct.

Observation 2: If there can be at most v errors at 2 sites s_A and s_B, and the first v combined signatures for s_A and s_B are identical, then sites s_A and s_B are both correct, otherwise, at least one of the sites is incorrect. □

Given that we have found a correct site, from Remark 1, one can identify the corrupted page copies if the first N combined signatures are computed. However, if we know that the maximum number of corrupted page copies is not large, we can compute and transmit less combined signature. In order to do this, we note the following remark.

Remark 3: [1] For any given $E_0, ..., E_{J-1} \in GF(2^b)$, where $1 \leq J \leq N$, the system of equations $\sum_{n=1}^{N} e_n \alpha^{jn} = E_j$ for $j = 0, ..., J-1$, has at most one solution for $(e_1, ..., e_N)$ with less than or equal to $\lfloor J/2 \rfloor$ non-zero entries. This solution can be obtained using a Reed-Solomon decoder [5]. □

In fact, only part of a Reed-Solomon decoder need to be utilized since we are interested in identifying the location of errors rather than in correcting the combined signatures for corrupted copies. To see Remark 3, Let us assume on the contrary that there are 2 different solutions $X = (e_1, e_2, ...e_N)$ and $X' = (e'_1, e'_2, ...e'_N)$, each having less than or equal to $\lfloor J/2 \rfloor$ non-zero entries.

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \vdots & \vdots & & \vdots \\ \alpha^{(J-1)} & \alpha^{2(J-1)} & \cdots & \alpha^{N(J-1)} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_N \end{pmatrix} = \begin{pmatrix} E_0 \\ E_1 \\ \vdots \\ E_{J-1} \end{pmatrix}$$

and

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \vdots & \vdots & & \vdots \\ \alpha^{(J-1)} & \alpha^{2(J-1)} & \cdots & \alpha^{N(J-1)} \end{pmatrix} \begin{pmatrix} e'_1 \\ e'_2 \\ \vdots \\ e'_N \end{pmatrix} = \begin{pmatrix} E_0 \\ E_1 \\ \vdots \\ E_{J-1} \end{pmatrix}$$

Let us subtract the second set of equations from the first set. We get

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha^1 & \alpha^2 & \cdots & \alpha^N \\ \vdots & \vdots & & \vdots \\ \alpha^{(J-1)} & \alpha^{2(J-1)} & \cdots & \alpha^{N(J-1)} \end{pmatrix} \begin{pmatrix} e_1 - e'_1 \\ e_2 - e'_2 \\ \vdots \\ e_N - e'_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Since $X = (e_1, e_2 ... e_N)$ and $X' = (e'_1, e'_2 ..., e'_N)$ both have less than $\lfloor J/2 \rfloor$ non-zero entries, the vector $X - X'$ has less than $2\lfloor J/2 \rfloor \leq J$ non-zero entries. Hence from Remark 2, we deduce that $X - X' = (0, 0, ..., 0)$, or the two sets of solutions are in fact identical. Therefore the solution is unique.

Observation 3: If we have the first J combined signatures of a correct site s_A and another site s_B, and we know that for s_B there can be at most $\lfloor J/2 \rfloor$ corrupted page copies, then there is a unique solution for the system of J equations as described in Remark 3, corresponding to the 2 set of combined signatures. The solution identifies the locations of all the corruptions in s_B. □

4. Distributed Algorithms

We make an observation that it is a reasonable assumption that a majority number of sites are correct. We propose a distributed algorithm (**Algorithm A**), as shown in Figure 1. Algorithm A makes three assumptions:

(1) Assumption A.
(2) $M > 2g$.
(3) the number of sites is a multiple of 2.

The third assumption is made only for the sake of simplicity in our presentation. We can readily relax it: If the number of sites is odd, we need only to partition the set SS into groups such that one and only one

ALGORITHM A:

Given: a set of sites SS.
Preprocessing Partition SS into groups of of size 2,
resulting in $GG = \{G_1, ...G_k\}$.
($G_i = \{\ s_{i1}, s_{i2}\ \}$ for $1 \leq i \leq k$)

The followings are operations at site s_{i1} of G_i.

Phase 1: Find a correct site

(1) Let $f' = \min\{N, f, 2h\}$
(2) SEND(REQUEST ($sig_{1,i2}, ...sig_{f',i2}$)) to s_{i2}.
(3) RECEIVE ($sig_{1,i2}, ...sig_{f',i2}$) from s_{i2} ....(R1)
(4) **If** $sig_{k,i1} = sig_{k,i2}$ *for* $1 \leq k \leq f'$
    Then msg_i = "Correct"
    Else msg_i = "Incorrect"
(5) SEND(msg_i) to s_{j1} of all other groups G_j.
(6) RECEIVE(msg_j) from s_{j1} of all other groups G_j.

Figure 1a: Phase 1 of Algorithm A

group has size 3, and the remaining groups has size 2. The algorithm proceeds in much the same way and the performance analysis is also similar.

A brief outline of the algorithm is first given:

1. Phase 1: partition the set of sites into groups of size 2 each, find a correct site by doing comparison of some combined signatures between the 2 sites in each group.

2. Phase 2: A set of groups FG will be found to contain corruptions. One site in each such group will be active, it first receives some combined signatures from a correct site found in Phase 1, in order to decide if itself is a correct site.

3. Phase 2 (cont.) The active site in each group determines if each site in the group is correct, if the other site is not correct, more of its combined signatures are requested if necessary. If both sites are incorrect, then more combined signatures are received from a correct site.

4. Phase 2 (cont.) The active site solves a set of equations to identify the error copies in each corrupted site in its group.

Next we describe the algorithm in more details. We shall refer to Figure 1 for the algorithm. In the figure, most of the computations are based on the local view

ALGORITHM A:

The followings are operations at site s_{i1} of G_i.

Phase 2: Find corrupted page copies

(1) ***Let FG be the set of groups G_j with msg_j ='Incorrect'*
(2) ***Let F be the number of groups in FG, i.e. $F = |FG|$.*
(3) ***Let $f'' = min\{h, f - F + 1\}$ and $f''' = min\{N, 2f''\}$.*

(4) **If** msg_i = 'Correct' *(The sites in this group are correct)*;
(5) **Then**
(6) If chosen by Protocol A, then
    SEND($sig_{1,c}, ...sig_{f'',c}$) to some sites,
(7) Wait for REQUEST messages and

    SEND signatures according to requests.
(8) **Else**
(9) *There is a site s_c which is correct*
    and chosen by Protocol A.
(10) RECEIVE ($sig_{1,c}, ...sig_{f'',c}$) from site s_c ....(R2)
(11) **If** $sig_{k,i1} = sig_{k,c}$ *for* $1 \leq k \leq f''$
(12) **Then** $d = i1$ *(Site s_{i1} is correct)* **Else** $d = c$

(13) **If** $sig_{k,i2} = sig_{k,c}$ *for* $1 \leq k \leq f''$
(14) **Then** *(Site s_{i2} is correct)*
(15) **If** $d \neq i1$ **Then** $d = i2$;

(16) **If** (s_{i2} is incorrect **or** $d = i2$) and $f''' > f'$
(17) **Then**
(18) SEND(REQUEST ($sig_{f'+1,s2}, ...sig_{f''',s2}$) to site s_{i2}
(19) RECEIVE ($sig_{f'+1,s2}, ...sig_{f''',s2}$)from site s_{s2}
    ....(R3)
(20) **If** *(both sites s_{i1} and s_{i2} are incorrect)*
(21) **Then**
(22) SEND(REQUEST ($sig_{f''+1,c}, ...sig_{f''',c}$) to site s_c
(23) RECEIVE ($sig_{f''+1,c}, ...sig_{f''',c}$) from site s_c
    ....(R4)

(24) **For** $m = i1$ and $m = i2$
(25) **If** it is not true that
    $sig_{k,m} = sig_{k,d}$ for $1 \leq k \leq f'''$
(26) **Then**
(27) Compute a vector $(e_{1,m}, ..., e_{N,m})$ of
    weight at most $min\{N, h, f - F + 1\}$
(28) as a solution to the system of equations

(29) $\sum_{n=1}^{N} e_{n,m}\alpha^{jn} = sig_{j,m} - sig_{j,d}$,
    where $j = 1, .., f'''$ (*)
(30) **End(For)**
    Note : ($e_{j,m}$ corresponds to $S(P_{i,m}) - S(P_{i,c})$)

(31) **....**For** m = 1 ,..., M
(32) **.... $C_m = \{P_{n,m} | 1 \leq n \leq N$ *and* $e_{n,m} \neq 0\}$
(33) **....**End(For)**
    (C_m is the set of corrupted page copies in site s_m.)

Figure 1b: Phase 2 of Algorithm A

of a site s_{i1} in a certain set of 2 sites. The lines marked with ** in Figure 1 are where global computation or global consideration of some kind is made.

In the first phase, we try to locate one or more correct site, as well as discover a number of sites which are correct and hence need not be considered in phase 2. We do this by partitioning the set of sites into groups of 2 sites each. The reason for comparing 2 sites is that by Observation 2, the comparison of 2 sites can discover correct sites if both the sites are correct, and that a set of 2 sites is the smallest possible set of sites to detect a correct file copy. More messaging is needed if we compare signatures in sets of more than 2 sites. Suppose we have sets of 3 sites, the number of sets of signatures to be sent in total is $\lceil \frac{2}{3}M \rceil$, which is greater than $\lceil \frac{1}{2}M \rceil$ for the case with sets of size 2.

Phase one is where Assumption A leads to a significant gain in minimizing the signature transmission compared to Consideration Z. With Consideration Z, in order to locate a correct site, one must find a majority of sites whose first k combined signatures are identical, where k is the number of possible errors at a site. In the worst case one would have to collect the combined signatures from every site before the majority is found. The number of signature transmissions is given by $(M-1)k$. With Assumption A, and given $M > 2g$, one can find a correct site by comparing 2 sites at a time, and hence in Phase one of Algorithm A, only $\frac{M}{2}k$ signature transmissions are necessary.

For each group in the partition, we compare a number of combined signatures. At this stage, we know that the maximal number of corrupted page copies at any site is $h \leq N$. The maximum possible number of differing page copies between 2 sites is given by $f' = \min\{N, f, 2h\}$. Hence we compare the first f' pairs of combined signatures. From Lemma 1 below, we shall discover at least one correct site.

In the second phase, we make use of the combined signatures of the correct site discovered in phase one to identify the corrupted pages in the corrupted sites. After Phase 1, we identify a set FG of groups of 2 sites, where each group has at least one corrupted site (refer to Figure 1). Suppose there are F groups in FG, it implies that at least F of the sites contain at least one corrupted page copy each. Then the maximal possible number of corrupted page copies at any site is given by $f'' = \min\{h, f - F + 1\}$. In order to identify f'' errors, from Observation 3, if $2f'' \leq N$, one needs to compare the first $2f''$ combined signatures with a correct site. If $2f'' > N$, from Observation 1, one needs to compare only the first N pairs of combined signatures. Therefore, we let $f''' = \min\{N, 2f''\}$ (Line (3)). The values of f'' and f''' will help to determine the number of signature transmission in the following steps.

In this phase, we shall apply a Protocol A which chooses a correct site for each of the groups in FG. For example, we may have unique site IDs for all sites and Protocol A may simply choose the correct site that has the smallest ID. Alternatively, Protocol A may choose different correct sites for different subsets of FG to minimize communication distances. Since we shall consider only the transmission of messages in our performance analysis, we shall not worry about the exact detail in Protocol A.

At each group $\{s_{i1}, s_{i2}\}$ in FG, one site, s_{i1}, is active. The first thing for s_{i1} to do is to find out if itself is a correct site, because if so, less number of combined signatures need to be received in a later stage. To do so, it receives the first f'' combined signatures from a correct site s_c. It compares its own first f'' combined signatures with these (Line (11)). By Observation 2, if the two sets of combined signatures are identical, then s_{i1} is correct. In this case, s_{i1} will act as the correct site s_d in discovering the corruptions of s_{i2}, and it would not need any more messages from s_c. Otherwise, s_{i1} is corrupted, in which case we try to find out if s_{i2} is correct in a similar way (Line(13)). If s_{i1} is corrupted but s_{i2} is correct, then it is better to let s_{i2} act as the correct site for error detection, since we have possibly received more combined signatures (the first f'' of them) from s_{i2} than from s_c. If s_{i2} is incorrect, then s_{i2} will send the $(f''+1)$-th to f'''-th combined signatures to s_{i1} for error detection (Lines (18-19)). If both s_{i1} and s_{i2} are found to be corrupted the role of correct site s_d is played by s_c.

If s_{i2} (s_c) is chosen as the correct site, then the remaining signatures in the first f''' combined messages, if any, will be collected from s_{i2} (s_c) (Lines (18-19) and Lines (22-23)). Finally, if s_m ($m = i1$ or $i2$) is corrupted, the solution to

$$\sum_{n=1}^{N} e_{n,m}\alpha^{jn} = sig_{j,m} - sig_{j,d}, \quad where\ j = 1,..,f''' \quad (*)$$

is computed (Lines (27-29)).

There are 2 possible cases, and the solution in each case gives a correct answer:

1. if $f''' = N$, then by Observation 1, we can find

the corrupted page copies;

2. if $f''' = 2f''$, since we know that the maximal number of corrupted pages in s_m is f'', therefore, by Observation 3, we can also find the corrupted page copies.

The following lemmas establish the correctness of the algorithm.

Lemma 1: In phase 1, we can always find a site s_c, and s_c contains no corruption.

Proof: First we see that at least one of the groups in GG contains 2 sites that are correct. This is because we assume that $M \geq 2g+1$, and if none of the groups in GG contains 2 sites that are correct, then g would be greater than $M/2$, a contradiction to our assumption.

For the two correct sites, the combined signatures will be identical. Hence we can always find a site s_c. From the definition of $sig_{k,i}$ and $sig_{k,j}$ we have

$$sig_{k,i} - sig_{k,j} = \sum_{n=1}^{N} (S(P_{n,i}) - S(P_{n,j}))\alpha^{kn}.$$

The vector $(S(P_{n,i}) - S(P_{n,j}))$ has weight at most $f' = \min\{N, f, 2h\}$. Since the combined signatures $sig_{k,i}$ and $sig_{k,j}$ for $j = 1, ..., f'$ agrees with each other, $\sum_{n=1}^{N}(S(P_{n,i}) - S(P_{n,j}))\alpha^{kn} = 0$. From Remark 2, we have $(S(P_{n,i}) - S(P_{n,j})) = 0$ for $n = 1, ..., N$. By Assumption A, two sites that contains corrupted pages will have different signatures. If the two sites have identical signatures, it means that the two sites are correct. Hence we conclude that site s_c has no corruption. □

Lemma 2: After phase 2, C_m contains the set of corrupted pages in site s_m.

Proof: In phase 2, f'' is the maximum number of corrupted pages in a site. First the correct site s_c sends its first f'' combined signatures to s_{i1} of all groups G_i which contains corrupted sites. From Remark 2, we can detect if site s_{i1} (s_{i2}) is corrupted by comparing its signatures with these corresponding ones from s_c. There are three possible outcomes:

1. If s_{i1} is not corrupted there is no need to receive further messages from s_c. In this case the first f''' combined signatures are gathered from s_{i2}. The role of a correct site is played by s_{i1}, i.e. s_d is set to s_{i1}.

2. If s_{i1} is corrupted and s_{i2} is correct, the first f''' combined signatures are gathered from s_{i2}. The role of a correct site is played by s_{i2}, i.e. s_d is set to s_{i2}.

3. If both s_{i1} and s_{i2} are corrupted, the first f''' combined signatures are gathered from s_c and s_{i2}. The role of a correct site is played by s_c, i.e. s_d is set to s_c.

After the message transmissions, we can compute the error locations in G_i. From the definition of $sig_{k,i}$ and $sig_{k,d}$ we have $sig_{k,i} - sig_{k,d} = \sum_{n=1}^{N}(S(P_{n,i}) - S(P_{n,d}))\alpha^{kn}$. Since s_d contains no corruptions, a non-zero value in $(S(P_{n,i}) - S(P_{n,d}))$ corresponds to a page corruption. If $N > 2f''$, then in Remark 3, $sig_{k,i} - sig_{k,d}$ corresponds to E_k, e_k corresponds to $(S(P_{n,i}) - S(P_{n,d}))$, and J corresponds to $2f''$. The equations (*) has a unique solution which corresponds to the set of corrupted pages. If $N \leq 2f''$, then Remark 1 implies that there is a unique solution. □

Performance Analysis

In phase one of Algorithm A the number of messages transmitted will be $\lceil \frac{M}{2} \rceil \min\{N, f, 2h\}$.

In the second phase, let F be the number of groups in FG. Consider the number of messages required for a group G_i of FG. More messages are to be received if site s_{i1} is not a correct site.

1. The total number of combined signatures transmitted from s_c for a RECEIVE at (R2) in Figure 1b is given by $f'' = \min\{h, f - F + 1\}$, and F sites would receive these messages.

2. Since g is the maximum number of failed sites, the number of groups in FG with 2 corrupted sites is bounded [2] from the above by $\lfloor g/2 \rfloor$. The total number of combined signatures transmitted from s_c for a RECEIVE at (R4) in Figure 1b is given by $f''' = \min\{N, 2f''\}$, and at most $\min\{F, \lfloor g/2 \rfloor\}$ sites would receive these messages.

3. The total number of combined signatures transmitted from s_{i2} for a RECEIVE at (R3) in Figure

[2] Note that we can refine the analysis by considering the number of groups in FG that contain 2 corrupted sites. Suppose this value is K. K is typically much smaller than F. This is because the chance of having 2 corrupted sites in a group in FG equals the probability that a site contains corruption, and for a reliable system this is small. If we take K into consideration in our analysis, we would arrive at better performance. We have not done so for the sake of simplicity.

1b is also given by f''', and at most F sites would receive these messages.

The total number of combined signature transmissions for (R1) and (R3) is bounded by

$$(\lceil \frac{M}{2} \rceil - F)f' + F \max\{f', f'''\}$$

where $f' = \min\{N, f, 2h\}$. The total number of combined signature transmissions for (R2) and (R4) is bounded by

$$f''(F - \min\{F, \lfloor g/2 \rfloor\}) + f''' \min\{F, \lfloor g/2 \rfloor\}.$$

To summarize, given that $f' = \min\{N, f, 2h\}$, $f'' = \min\{h, f - F + 1\}$, and $f''' = \min\{N, 2f''\}$, the number of combined signature transmissions for Algorithm A is bounded by

$$\gamma^A(M, N, f, g, h, F) = (\left\lceil \frac{M}{2} \right\rceil - F)f' + F \max\{f', f'''\} + f''(F - \min\{F, \lfloor \frac{g}{2} \rfloor\}) + f''' \min\{F, \lfloor \frac{g}{2} \rfloor\}.$$

If there is no failure, then the number of combined signature transmissions is given by

$$\gamma_0^A(M, N, f, h, F) = \left\lceil \frac{M}{2} \right\rceil \min\{N, f, 2h\}$$

We shall refer to the algorithm in [1] as Algorithm Z. Let $\beta(M, N, f)$ be the maximum number of signature transmission required for Algorithm Z. We show the number of signature transmission required for Algorithms A and Z under different system parameters of N, M, f, g and F in Figures 2 and 3. In the figures, $\beta(M, N, f)$ is denoted by $Beta(w)$, $\beta(M, N, f)$ for the case of no failure is denoted by $Beta(0)$, $\gamma^A(M, N, f, g, h, F)$ is denoted by $Gamma(A)$.

The value $\gamma_0^A(M, N, f, h, F)$ for the case of no failure ($F = 0$) is a lower bound on the value of $\gamma^A(M, N, f, g, h, F)$. From Figure 2, it is seen that the value of $\gamma^A(M, N, f, g, h, F)$ is furthest away from $\gamma_0^A(M, N, f, h, F)$ for a relatively unreliable system, where the number of possible failing sites is almost half the number of sites (Figure 2(a)). The difference becomes smaller as the number of possible failing sites is decreased.

In Figure 2, the value of f depends only on g. In Figure 3, the value of f depends only on both M and N. We can see from the figures that Algorithm A performs well in that it requires less signature messages than Algorithm Z.

Our final set of figures illustrates the importance of parameter h, the maximum number of corrupted page

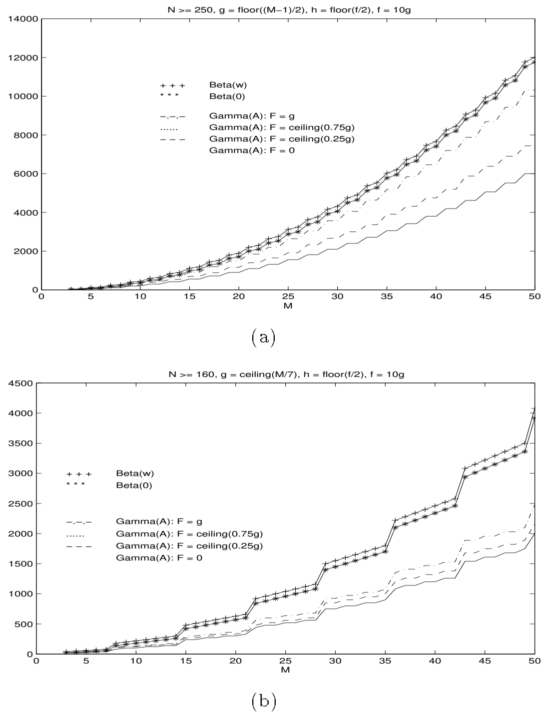


Figure 2: Number of signature transmissions for Algorithms A and Z

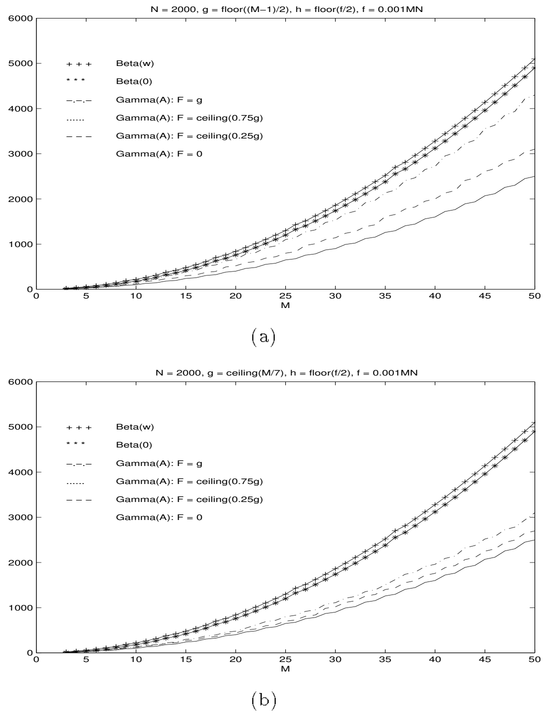
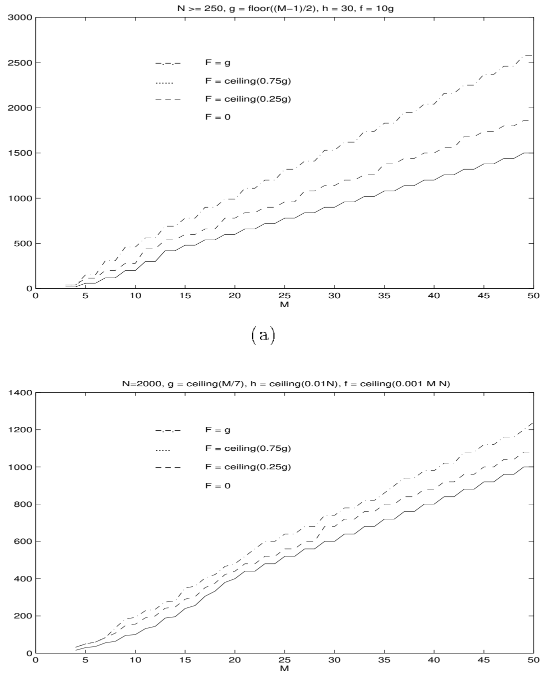


Figure 3: Cases of f varying with M and N



(a)

(b)

Figure 4: Effects of reasonable values of h

copies at a site. If this is not known, h may be assume to be $\min\{N, f\}$. However, it is not likely that all pages of a file are corrupted. In the figures, we assume some reasonable values of h, such as 10% of the total number of pages. Compared with Figures 2 and 3, we see that the number of required signature transmission are reduced significantly.

5. Conclusion

By making the assumption that different sites are unlikely to contain corrupted page copies of a page in a replicated file that have identical signatures, we propose an algorithm for locating corruptions which has better performance than previous work with less number of signature transmission. We propose a finer system model in which the maximum number of corrupted copies in a site is considered, and we show that it can lead to significant enhancement in the performance of the proposed algorithm.

Acknowledgements

We thank the anonymous referees whose comments have helped to improve the paper tremendously. The first auther was supported by the RGC (the Hong Kong Research Grants Council) grant UGC REF.CUHK 495/95E.

References

[1] K. Abdel-Ghaffar and A. El Abbadi. Efficient detection of corrupted pages in a replicated file. In *12th ACM Symposium on Principles on Distributed Computing*, pages 219–227, 1993.

[2] K. Abdel-Ghaffar and A. El Abbadi. An optimal strategy for comparing file copies. *IEEE Transactions on Parallel and Distributed Systems*, pages 87–93, Jan 1994.

[3] A. Abdennadher and J. Metzner. Use of guass-jordan matrix reduction in the reconstruction of a correct file copy from erroneous copies. *Proc. 27th Hawaii International Conference on System Sciences*, pages 330–340, Jan 1994.

[4] D. Barbara, H. Garcia-Molina, and B. Feijoo. Exploiting symmetries for low-cost comparison of file copies. In *Proc. Int. Conf. Distributed Comput. Syst.*, pages 471–479, June 1988.

[5] R. Blahut. *Theory and Practice of Error Control Codes*. Addison Wesley, Reading, MA, 1984.

[6] W. Fuchs, K. Wu, and A. J.A. Low-cost comparison and diagnosis of large remotely located files. In *Proc. Symp. Reliability of Distributed Software and Database Systems*, pages 67–73, Jan 1986.

[7] J. Metzner. A parity structure for large remotely located replicated data files. *IEEE Transactions on Computers*, pages C–32:727–730, Aug 1983.

[8] J. Metzner. Reliable and efficient broadcast of files to a group of locally interconnected stations. *Proc. GLOBECOM '86*, pages 1762–1767, Dec 1986.

[9] J. Metzner. Efficient replicated remote file comparison. *IEEE Transactions on Computers*, pages C–40(5):651–660, May 1991.

[10] J. Metzner and Kapturowski. A general decoding technique applicable to replicated file disagreement location and concatenated code decoding. *IEEE Transactions on Inform. Theory*, pages IT–36:911–917, July 1990.