Journal of Supercomputing, 5, 1–34 (1997)
© 1997 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Locating Corruptions in a Replicated File in a Distributed Environment

ADA WAICHEE FU adafu@cse.cuhk.edu.hk
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong

and

SIU-CHEUNG CHAU chau@cs.uleth.ca
Dept. of Mathematics and Computer Science, University of Lethbridge, Lethbridge, Alberta, Canada, T1K 3M4.

Received November 25, 1995; Revised March 30, 1996

Editor: Hamid R. Arabnia

Abstract. When a data file is replicated at more than one site, it is of interest to detect 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 is at most $\lceil M/2 \rceil - 1$, where M is the number of sites. This is a pessimistic assumption which is unrealistic. In this paper, this assumption is replaced by another assumption which is shown to be reasonable. Based on this assumption, and based on a finer model of the system, three distributed algorithms are derived, which can either improve the performance or provide more tolerance to corruptions compared to previous methods. As in some previous work, the amount of signature transmission in the algorithms varies according to the number and patterns of page copy corruptions that actually occur, and two of the algorithms achieve the optimal amount of signature transmission when no failure occurs.

Keywords: replicated file, fault tolerance, file comparison, coding theory, performance analysis, message transmission, signatures.

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. Since each page in a file may be large, a more concise representation, called a *page signature* is computed based on the information stored in each page. The size of a page signature is typically much smaller than the size of the page, though some information may be lost in the compression. From the page signatures, *combined signatures* are computed and transmitted among sites. The corruptions are detected by performing some computations on the 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 caters for the worst case scenario in which 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. It is noted 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, it is argued 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. Thus here it is assumed that 2 corrupted page copies do not have the same signature. With the new assumption, and with a finer model of the system, 3 distributed algorithms are derived that either need to transmit a smaller number of signatures or have greater tolerance of corruption. Two of the algorithms are shown to achieve the minimal number of signature transmission when no failure occurs. One of the algorithms (Algorithm GF_SIG_A) has been presented in [6].

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 algorithms that follow. Section 4 presents the distributed algorithms. Section 5 is a performance analysis and comparison with previous work. Section 6 gives a lower bound of the problem in non-faulty case and optimality of the algorithms with respect to this. Section 7 relaxes some restrictions. Concluding remarks are given in Section 8.

2. Problem Definition

Consider the case where a file is replicated among a set of M sites $SS = \{s_1, ..., s_M\}$. 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. In this paper decentralized control is assumed 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, it is assumed that the total number of corrupted page copies will not exceed f. Furthermore, it is assumed that each site contains at most h corrupted page copies, and at most g sites have corruption. If these values are not known, then h is set to $\min\{N, f\}$, and g is set to a greatest possible value.

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 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 [8] 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.

The number of page signatures may still be large. To further reduce the amount of messages that need to be sent around, each site could compute a number of *combined signatures* from the page signatures. For example, the combined signature can be a bit-wise exclusive OR of a number of page signatures. Hence the number of combined signatures will be smaller than the 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 [9, 11]. In particular, a method using Reed-Solomon codes [5] has been suggested in [11] for problems with only 2 sites. [1] extended the method of Reed-Solomon codes and solves the problem for M replicated sites, where $M \geq 3$.

Table 1 summarizes the characteristics of a number of previous work for this problem. In Table 1, N is the number of pages in the file, a1 is the assumption that an uncorrupted copy resides in a known site, and a2 is the assumption that N is a power of 2.

Algorithm	# copies	a1	a2	max # failures	# signatures transmitted
Met86,91[9][10]	2	Yes	Yes	f	$f[(log_2 N) + 1]$
FWA86[7]	2	Yes	No	1	$(log_2 N) + 1$
BGMF88[4]	2	Yes	No	2	$(log_2 N)[log_2 N + 1]/2 + 1$
MK90[11]	2	Yes	No	f	$f + 1$
AGE94[3]	2	Yes	No	f	$min\{N, 2f\}$
AGE93[1] (centralized)	M	No	No	f	$(M - 2)min\{N, f\} + min\{N, 2f\}$
AGE93[1] (decentralized)	M	No	No	f	$(M - \lfloor M/(2f + 1)\rfloor - 1)\min\{N, f\}$ $+ \min\{N, 2f\}$

Table 1: results of previous work

The following 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 2 : Notations

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

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 $M < 2f+1$, and a majority number of sites has failed at the same page, then a majority of the page signatures (the corrupted ones) of this page can be identical and there is no way to tell whether the corresponding copies are corrupted.

Consider 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. It can be concluded 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 it is shown that such an assumption is reasonable. In particular, This assumption is supported 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 such as [3, 1, 11, 10]. 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. $Prob_A$ and $Prob_B$ will be shown to be no greater than $Prob_C$, and hence $Prob_A$ and $Prob_B$ should also be negligible.

The 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, it is assumed 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 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 note that $Prob_A$ is comparable to acceptable values of $Prob_C$. For example, for the algorithm given in [8], 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. For a 40-bit parity parity sequence suggested in the paper, this probability will be about 10^{-12}. The value of b is typically large compared to c, for example, in [8], 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.

This paper thus put forward the proposal that the probability of 2 corrupted pages having the same page signature is negligible. It will be seen that this relaxation leads to algorithms with better performance or fault-tolerance. The assumption in this paper is the following:

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

Assumption A is based on a more basic assumption:

Assumption B: *site failures are independent.*

Assumption B 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.

3. Properties of Combined Signatures

We shall adopt the combined signature techniques as in [11, 3, 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.

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 $1 \le j \le 2^b - 1$, 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 $k \ge 1$, as the **first** k combined signatures.

To help explain the algorithm in the next section, a few properties of the combined signatures are first described. In the algorithm two sets of combined signatures from two different sites s_A and s_B are compared. In a special case, suppose the first N combined signatures in each site are computed, and it is known 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

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

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

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

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

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

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

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

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 it is assumed that $N < 2^b$, 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 site 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:*

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

to identify the location of all corruptions in s_A.

Observation 1 provides a way to detect errors at a corrupt site given a correct site. However, in the beginning we do not know any correct site, a correct site must first be discovered. Here a property of the combined signatures allows such a discovery without the need of sending too many combined signatures. It will be shown that if there are at most v errors located at a set of 2 sites, one needs 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

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ \alpha^1 & \alpha^2 & \dots & \alpha^N \\ : & : & & : \\ : & : & & : \\ \alpha^{v-1} & \alpha^{2(v-1)} & \dots & \alpha^{N(v-1)} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ : \\ : \\ e_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ : \\ : \\ 0 \end{pmatrix} \tag{3}$$

Remark 2 is based on the following argument. 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.

$$\begin{pmatrix} 1 & 1 & \ldots & 1 \\ \alpha^{(i_1)i_1} & \alpha^{(i_2)i_1} & \ldots & \alpha^{(i_t)i_1} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ \alpha^{(i_1)(i_{t-1})} & \alpha^{(i_2)(i_{t-1})} & \ldots & \alpha^{(i_t)(i_{t-1})} \end{pmatrix} \begin{pmatrix} e_{i_1} \\ e_{i_2} \\ \vdots \\ \vdots \\ e_{i_t} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \tag{4}$$

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 (3). The left-most matrix in (4) 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 set of equations in Remark 2 can be written as

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

Therefore, if the first v combined signatures for s_A and s_B are compared and it is found 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 it is assumed (Assumption A) 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.*

Given that a correct site is found, from Remark 1, one can identify the corrupted page copies if the first N combined signatures are computed. However, if it is known that the maximum number of corrupted page copies is not large, one can compute and transmit fewer combined signatures. In order to do this, 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 fewer 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 it is of interest to identify the location of errors rather than in correcting the combined signatures for corrupted copies. Remark 3 is based on the following argument. First 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 fewer than or equal to $\lfloor J/2 \rfloor$ non-zero entries.

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

and

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

Next subtract the second set of equations from the first set:

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

Since $X = (e_1, e_2...e_N)$ and $X' = (e'_1, e'_2..., e'_N)$ both have fewer than $\lfloor J/2 \rfloor$ non-zero entries, the vector $X - X'$ has fewer 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 the first J combined signatures of a correct site s_A and another site s_B are given, and it is known 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

The decentralized algorithm in [1] is considered since it achieves better performance. Call this algorithm **Algorithm GF_SIG**. Algorithm GF_SIG works for $M > 2g$, since a majority number of sites will be correct and contain identical combined signatures. However, since the algorithms are based on a very pessimistic consideration, they may be using more message transmissions than necessary. In the first proposed algorithm (Algorithm GF_SIG_A), the amount of messaging is reduced mainly due to replacing Assumption Z and Consideration Z by Assumption A.

The second proposed algorithm (Algorithm GF_SIG_B) makes use of a more refined model, which considers the parameter of h, the maximum number of corrupted pages at a site. A further improvement on the number of message transmission is achieved, given reasonable values of h.

Finally Algorithm GF_SIG is modified to handle the case of $M > g + 2$ given Assumption A. The amount of message transmission is the same but the tolerance of corruptions is greater than Algorithm GF_SIG. Note that Algorithm GF_SIG cannot handle the case where $M \leq 2g$ because in such cases there can be a majority of sites that are corrupted, and Algorithm GF_SIG will not be able to find a correct site.

4.1. Algorithm GF_SIG_A

In this subsection, a distributed algorithm called **Algorithm GF_SIG_A** is proposed. The parameter of h is not considered. Algorithm GF_SIG_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 presentation. This can be readily relaxed : If the number of sites is odd, partition the set SS into groups such that one and only one 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.

Consider two different cases:

Case 1: $N \leq f$

The sites are first partitioned into groups of 2 sites, and the N page signatures between the 2 sites in each group are compared. Any two page signatures that do not agree indicates one or two corrupted copies. At the end of the comparison of all groups, one must be able to locate at least one correct site since there must be at least one group that contains two correct sites (see Lemma 2 at the end of this subsection). The amount of signature transmission is given by $\lceil M/2 \rceil N$. Let FG be the set of groups that are found to contain some corruption. Suppose there are F groups in FG. In the second phase, the correct site sends its page copies to locate the corruptions in each of the groups in FG. The maximum number of signatures sent for this purpose is bounded by FN. Hence the total number of signature transmission is $(\lceil M/2 \rceil + F)N$.

Case 2: $N > f$

For $N > f$, the procedure is shown in Figure 1. Most of the computations in Figure 1 are based on the local view of a site s_{i1}. Lines (1)-(2), (19)-(21), and (36)-(38) in Figure 1 are where global computation or global consideration of some kind is made.

For Case 2, Algorithm GF_SIG_A is made up of 2 phases. Phase 1 will either discover that all sites are correct or locate at least one correct site. Phase 2 makes use of the combined signatures of the correct site to detect corruptions.

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 comparison of some combined signatures between the 2 sites in each group.

ALGORITHM GF_SIG_A:

Given: a set of sites SS.
Preprocessing Partition SS into groups of size 2.
Let the partitioning be $GG = \{G_1, ...G_p\}$. ($G_i = \{ s_{i1}, s_{i2} \}$ for $1 \leq i \leq p$)

The followings are operations at site s_{i1} of G_i.
Phase 1: Find a correct site

(1) SEND(REQUEST ($sig_{1,i2}, ...sig_{f,i2}$)) to s_{i2}.
(3) RECEIVE ($sig_{1,i2}, ...sig_{f,i2}$) from s_{i2}
(4) **If** $sig_{k,i1} = sig_{k,i2}$ *for* $1 \leq k \leq f$ **Then** $message_i$ = "Correct"
Else $message_i$ = "Incorrect"
(5) SEND($message_i$) to s_{j1} of all other groups G_j.
(6) RECEIVE($message_j$) from s_{j1} of all other groups G_j.

Figure 1a: Phase 1 of distributed algorithm for detecting corruption

2. Phase 2: A set of groups FG will be found to contain corruptions. One site s_{i1} in each group G_i in FG will be active, it receives the first f combined signatures from a correct site found in Phase 1.

3. Phase 2 (cont.): s_{i1} solves a set of equation in order to detect corruptions if the sites of this group has fewer than $f/2$ errors each.

4. Phase 2 (cont.): After the above step, at most one site will be discovered to contain more than $f/2$ corrupted page copies. $2f$ combined signatures are requested from a correct site for the error detection of this site.

Next the algorithm is described in more details.

Phase 1 tries to locate one or more correct sites, as well as discover a number of sites which are correct and hence need not be considered in Phase 2. This is done 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 signatures in groups of more than 2 sites are compared. Suppose groups are 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 groups of size 2.[1] It will be seen in Section 6 that with this arrangement,

ALGORITHM GF_SIG_A:

The followings are operations at site s_{i1} of G_i.
Phase 2: Find corrupted page copies

**....Let FG be the set of groups G_j with $message_j$ = "Incorrect"
**.....Let F be the number of groups in FG.
If$message_i$ = "Correct" *(The sites in this group are correct)*;
Then
  If chosen by Protocol A, SEND the first f combined signatures to some sites
  Wait for REQUEST messages and SEND signatures according to requests.
Else
  There is a site s_c which is correct and chosen by Protocol A for G_i.
  RECEIVE $(sig_{1,c}, ...sig_{f,c})$ from site s_c
  If $sig_{k,i2} = sig_{k,c}$ *for* $1 \le k \le f$ **Then** $d = i2$ *(Site s_{i2} is correct)*
  If $sig_{k,i1} = sig_{k,c}$ for $1 \le k \le f$ **Then** $d = i1$ *(Site s_{i1} is correct)*
          Else $d = c$ *(both sites s_{i1} and s_{i2} are incorrect)*
  For $m = i1$ and $m = i2$
    If it is not true that $sig_{k,m} = sig_{k,d}$ for $1 \le k \le f$
    Then *(The site is corrupted)*
      Compute a vector $(e_{1,m}, ..., e_{N,m})$ of at most $min\{N, f/2\}$
      non-zero elements as a solution to the system of equations
      $\sum_{n=1}^{N} e_{n,m}\alpha^{jn} = sig_{j,m} - sig_{j,d}$, where $j = 1, .., f$
If $f = N$ or (each site has $\le f/2$ errors), then all corrupted pages are found
Else at most one site s_w has more than $f/2$ errors, s_w has no solution in the above.
Let $f'' = \min\{N, 2(f - G)\}$ where G is the number of corrupted page copies found so far.
If s_w exists and is in this group
Then
  If s_{i1} is correct **Then** $s_d = s_{i1}$ **Else** $s_d = s_c$
  If $s_w = s_{i2}$
  Then
    SEND(REQUEST $(sig_{f+1,i2}, ...sig_{f'',i2})$) to site s_{i2}
    RECEIVE $(sig_{f+1,i2}, ...sig_{f'',i2})$ from site s_{i2}
  If $s_d \ne s_{i1}$
  Then
    SEND(REQUEST $(sig_{f+1,d}, ...sig_{f'',d})$) to site s_d
    RECEIVE $(sig_{f+1,d}, ...sig_{f'',d})$ from site s_d
  Compute a vector $(e_{1,w}, ..., e_{N,w})$ with at most f non-zero elements
      as a solution to the system of equations
      $\sum_{n=1}^{N} e_{n,w}\alpha^{jn} = sig_{j,w} - sig_{j,d}$, where $j = 1, .., f''$
.... For** m = 1 ,..., M
**.... $C_m = \{P_{n,m} | 1 \le n \le N \text{ and } e_{n,m} \ne 0\}$
**.... *(Note: C_m is the set of corrupted pages in site s_m.)*

Figure 1b: Phase 2 of Algorithm GF_SIG_A a minimum amount of combined signature transmission is achieved if no failures occur, which is a desirable property given a relatively reliable system.

Phase 1 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, one can find a correct site by comparing 2 sites at a time, and hence in Phase 1 of Algorithm GF_SIG_A, only $\frac{M}{2}k$ signature transmissions are necessary.

In Phase 1, for each group of 2 sites, a number of combined signatures are compared. The maximum possible number of differing page copies between 2 sites is given by f. Hence the first f pairs of combined signatures are compared. By Observation 2, if the values of the two sets of combined signatures $(sig_{1,i}, ...sig_{f,i})$ and $(sig_{1,j}, ...sig_{f,j})$ are the same, then it means that both sites s_i and s_j have no corruption. Otherwise, one or both contains corruption. it can be sured that at least one group contains correct sites (see Lemma 2 in the following). The groups inform each other whether they contain corruptions by sending messages "Correct" or "Incorrect" to each other.

In Phase 2, for each group G_i of FG, the site s_{i1} is active, and its operations are shown in Figure 1b. In this phase, a Protocol A is applied which chooses a correct site found in Phase 1 for each of the groups in FG. For example, there may be 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 this paper considers only the transmission of messages in the performance analysis, the exact detail in Protocol A is not of concern.

The first mission for s_{i1} is to receive the first f combined signatures from the chosen correct site s_c. There can be at most f corruptions at a site, therefore, by Observation 2, these f combined signatures can be used to determined whether s_{i1} or s_{i2} are correct sites (Lines 10 to 12 of Algorithm GF_SIG_A).

The next step is to compute a solution with at most $f/2$ non zero elements to the set of the first f signature equations for each corrupted site in G_i (Lines (16)-(18) of Algorithm GF_SIG_A). There are two possible cases,

1. In the first case, each site has fewer than $f/2$ errors, since there are in total at most f errors, then by Observation 3, there will be a unique solution for the signature equations for each group in FG, and the solutions solve the problem.

2. In the second case, not every site has fewer than $f/2$ then there is at most one site which has more than $f/2$ errors, and at this site, if it exists, there is no solution with at most $f/2$ non-zero elements in the above computation. Let this site be s_w if it exists.

Continue with the steps for Case (2) above. For this case, corruptions at all sites other than s_w have been detected. For s_w, more correct signatures will be necessary for discovering the corrupted copies. Suppose G errors have been discovered at sites other than s_w, then the maximum number of errors at s_w is bounded by $f-G$. Let $f'' = \min\{N, 2(f-G)\}$. By Observations 1 and 3, the first f'' combined signatures of s_w is compared with a correct site to solve the corresponding set of equation. If s_w is s_{i2} of group G_i, since the first f of the combined signatures of both s_w and a correct site is known, site s_{i1} will ask for the next $f+1$ to f'' combined signatures from s_{i2}. In this case, if s_{i1} is not a correct site, it would need to request for the f to f'' combined signatures from s_c for the error detection. If s_w is s_{i1}, then it needs only collect the next $(f+1)$ to f'' combined signatures from a correct site s_c. Finally, a solution to the set of equations at Line (35) of Algorithm GF_SIG_A in Figure 1b will give a solution for s_w. Since this is the only site remaining unsolved, error detection is now completed. The following lemma is thus shown:

LEMMA 1 : *Algorithm GF_SIG_A computes the locations of corruption under the given assumptions.*

The following lemma establishes the correctness of Phase 1.

LEMMA 2 : *Phase 1 of Algorithm GF_SIG_A can always find a site s_c, and s_c contains no corruption.*

Proof: First note that at least one of the groups in GG contains 2 sites that are correct. This is because of the assumption $M \geq 2g+1$, 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 the assumption.

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

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

The vector $(S(P_{n,i}) - S(P_{n,j}))$ has weight at most f. 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, $(S(P_{n,i}) - S(P_{n,j})) = 0$ for $n = 1, ..., N$. By Assumption A, two sites that contain corrupted pages will have different signatures. If the two sites have identical signatures, it means that the two sites are correct. Hence one concludes that site s_c has no corruption. □

4.2. Algorithm GF_SIG_B: Considering bound on corruptions per site

The next proposed algorithm is called **Algorithm GF_SIG_B**. In this algorithm a finer model than that of Algorithm GF_SIG_A is adopted. In particular, h, the maximum number of corrupted page copies at a site at file comparison is considered. Algorithm GF_SIG_B is shown in Figure 2. As in Algorithm GF_SIG_A, three assumptions are made:

(1) Assumption A.

(2) $M > 2g$.

(3) the number of sites is a multiple of 2.

As in Algorithm GF_SIG_A, the third assumption is made only for the sake of simplicity in the presentation. 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 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.

ALGORITHM GF_SIG_B:

Given: a set of sites SS.
Preprocessing Partition SS into groups of size 2.
Let the partitioning be $GG = \{G_1, ...G_p\}$. ($G_i = \{\ s_{i1}, s_{i2}\ \}$ for $1 \leq i \leq p$)

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}
(4) **If** $sig_{k,i1} = sig_{k,i2}$ *for* $1 \leq k \leq f'$ **Then** $message_i$ = "Correct"
Else $message_i$ = "Incorrect"
(5) SEND($message_i$) to s_{j1} of all other groups G_j.
(6) RECEIVE($message_j$) from s_{j1} of all other groups G_j.

Figure 2a: Phase 1 of Algorithm GF_SIG_B

Next the algorithm is described in more details. Algorithm GF_SIG_B is shown in Figure 2. In the figure, most of the computations are based on the local view of a site s_{i1} in a certain set of 2 sites. The lines (1)-(3) and (31)-(33) in Figure 2 are where global computation or global consideration of some kind is made.

Phase 1 tries to locate one or more correct sites, as well as discover a number of sites which are correct and hence need not be considered in Phase 2. As in Algorithm GF_SIG_A, this is done by partitioning the set of sites into groups of 2 sites each. The reasons for this are the same as before. For each group in the partition, a number of combined signatures are compared. At this stage, the maximal number of corrupted page copies at any site is $h \leq N$ is known. The maximum possible number of differing page copies between 2 sites is given by $f' = \min\{N, f, 2h\}$. Hence the first f' pairs of combined signatures are compared. Using the arguments in Algorithm GF_SIG_A, at least one correct site will be discovered.

Phase 2 makes 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, a set FG of groups of 2 sites is identified, where each such group has at least one corrupted site (refer to Figure 2). Suppose there are F groups in FG, it implies that at least F of the sites contain at least one corrupted page copy each.

ALGORITHM GF_SIG_B:

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

Phase 2: Find corrupted page copies

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

If $message_i$ = "Correct" *(The sites in this group are correct)*;
Then
    If chosen by Protocol A, SEND($sig_{1,c}, ...sig_{f'',c}$) to some sites and Wait
    for REQUEST messages and SEND signatures according to requests.
Else
    There is a site s_c which is correct and chosen by Protocol A.
    RECEIVE ($sig_{1,c}, ...sig_{f'',c}$) from site s_c
    If $sig_{k,i1} = sig_{k,c}$ *for* $1 \leq k \leq f''$
    Then $d = i1$ *(Site s_{i1} is correct)* **Else** $d = c$
    If $sig_{k,i2} = sig_{k,c}$ *for* $1 \leq k \leq f''$
    Then *(Site s_{i2} is correct)*
        If $d \neq i1$ **Then** $d = i2$;
    If (s_{i2} is incorrect **or** $d = i2$) and $f''' > f'$
    Then
        SEND(REQUEST ($sig_{f'+1,s2}, ...sig_{f''',s2}$)) to site s_{i2}
        RECEIVE ($sig_{f'+1,s2}, ...sig_{f''',s2}$) from site s_{i2}
    If *(both sites s_{i1} and s_{i2} are incorrect)*
    Then
        SEND(REQUEST ($sig_{f''+1,c}, ...sig_{f''',c}$)) to site s_c
        RECEIVE ($sig_{f''+1,c}, ...sig_{f''',c}$) from site s_c

    For $m = i1$ and $m = i2$
        If it is not true that $sig_{k,m} = sig_{k,d}$ for $1 \leq k \leq f'''$
        Then
            Compute a vector $(e_{1,m}, ..., e_{N,m})$ of weight at most
            $min\{N, h, f - F + 1\}$ as a solution to the system of equations
                $\sum_{n=1}^{N} e_{n,m}\alpha^{jn} = sig_{j,m} - sig_{j,d}$, where $j = 1, .., f'''$
    End(For) *Note : ($e_{j,m}$ corresponds to $S(P_{i,m}) - S(P_{i,c})$)*

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

Figure 2b: Phase 2 of Algorithm GF_SIG_B

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, let $f''' = \min\{N, 2f''\}$ (Line (3) of Algorithm GF_SIG_B). The values of f'' and f''' will help to determine the number of signature transmission in the following steps.

As in Algorithm GF_SIG_A, Protocol A is applied to choose a correct site for each of the groups in FG. At each group $\{s_{i1}, s_{i2}\}$ in FG, one site, s_{i1}, is active. The first task for s_{i1} is to find out if itself is a correct site, because if so, fewer 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) of Algorithm GF_SIG_B). 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 the way to find out if s_{i2} is correct is similar (Line(13) of Algorithm GF_SIG_B). 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 possibly more combined signatures (the first f'' of them) are received 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) of Algorithm GF_SIG_B). 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) of Algorithm GF_SIG_B).

Finally, if s_m ($m = i1$ or $i2$) is corrupted, the solution to

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

is computed (Lines (27)-(29) of Algorithm GF_SIG_B).

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

1. if $f''' = N$, then by Observation 1, the corrupted page copies can be found;
2. if $f''' = 2f''$, since the maximal number of corrupted pages in s_m is f'', therefore, by Observation 3, the corrupted page copies can also be found.

4.3. Algorithm GF_SIG_C: Considering bound on number of corrupted sites

Given Assumption A, one can assume that $M > g + 2$, where g is the maximum number of sites that contain corruptions. If $M > g + 2$, then at least 2 sites are correct, and the algorithm in [1] can be slightly modified to detect all corruptions for this case: In finding a correct site, instead of finding a majority of sites that agree with each other, find a set of 2 sites that agree with each other. In the worst case, the resulting amount of messaging is the same as the original algorithm but now a much greater number of possible errors can be tolerated. The resulting algorithm is called **Algorithm GF_SIG_C**.

5. Performance Analysis

In this section, the performance of Algorithm GF_SIG_A and Algorithm GF_SIG_B is analyzed. The performance among the two algorithms and Algorithm GF_SIG are also compared. Both proposed Algorithms are shown to perform better than Algorithm GF_SIG in that fewer number of signature transmissions is necessary. The choice between Algorithm GF_SIG_A and Algorithm GF_SIG_B are also discussed.

5.1. Performance of Algorithm GF_SIG_A

In Phase 1 of Algorithm GF_SIG_A, the number of messages transmitted will be $\lceil \frac{M}{2} \rceil f$.

In Phase 2, 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 required if site s_{i1} is not a correct site.

1. The total number of combined signatures transmitted from s_c for a RECEIVE at Line (9) of Algorithm GF_SIG_A in Figure 1b is given by f, and at most F sites would receive these messages. Since f is the maximum number of failures, F is at most f.

2. In the worst case no error may be found at sites other than s_k. The total number of combined signatures transmitted from s_c for a RECEIVE at Line (32) of Algorithm GF_SIG_A in Figure 1b is bounded by $\min\{N - f, f\}$, and at most one site would receive these messages.

3. The total number of combined signatures transmitted from s_{i2} for a RECEIVE at Line (28) of Algorithm GF_SIG_A in Figure 1b is bounded by $\min\{N-f, f\}$, and at most one site would receive these messages.

Together with the consideration of the cases for $N \leq f$, the total number of combined signature transmissions is bounded by

$$\gamma^A(M,N,f) = \left(\lceil \frac{M}{2} \rceil + f\right) \min\{N,f\} + 2\min\{\max\{N-f,0\},f\}$$

A tighter bound is to consider the value of F which is defined in the above. The bound is

$$\gamma_t^A(M,N,f) = \left(\lceil \frac{M}{2} \rceil + F\right) \min\{N,f\} + 2\min\{\max\{N-f,0\},f\}$$

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

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

5.2. Performance of Algorithm GF_SIG_B

In Phase 1 of Algorithm GF_SIG_B the number of messages transmitted will be $\frac{M}{2}\min\{N,f,2h\}$.

In Phase 2, 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 Line (10) of Algorithm GF_SIG_B in Figure 2b 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 from the above by $\lfloor g/2 \rfloor$. The total number of combined signatures transmitted from s_c for a RECEIVE at Line (23) of Algorithm GF_SIG_B in Figure 2b is given by $f''' = \min\{N, 2f''\}$, and at most $\min\{F, \lfloor g/2 \rfloor\}$ sites would receive these messages.[2]

3. The total number of combined signatures transmitted from s_{i2} for a RECEIVE at Line (19) of Algorithm GF_SIG_B in Figure 2b is also given by f''', and at most F sites would receive these messages.

The total number of combined signature transmissions for Lines (3) and (19) of Algorithm GF_SIG_B is bounded by

$$(\tfrac{M}{2} - F)f' + F\max\{f', f'''\}$$

where $f' = \min\{N, f, 2h\}$. The total number of combined signature transmissions for Lines (10) and (23) in Algorithm GF_SIG_B 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 GF_SIG_B is bounded by

$$\begin{aligned} & \gamma^B(M, N, f, g, h, F) \\ = \; & (\frac{M}{2} - F)f' + \max\{f', f'''\} + f''(F - \min\{F, \lfloor \frac{g}{2} \rfloor\}) + f''' \min\{F, \lfloor \frac{g}{2} \rfloor\}. \end{aligned}$$

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

$$\gamma_0^B(M, N, f, h, F) = \frac{M}{2} \min\{N, f, 2h\}$$

5.3. Performance Comparisons: Cases of No Failure

For reliable systems, it is likely that no failure has occurred at most file comparisons. Therefore first consider the cases of no failure.

Compare the performance of Algorithm GF_SIG_A and Algorithm GF_SIG_B with the bounds achieved by [1]. Recall that the more efficient decentralized algorithm in [1] is called Algorithm GF_SIG. The maximum number of signature transmission it requires is given by

$$\beta_w(M, N, f) = (M - \lfloor M/(2f + 1) \rfloor - 1) \min\{N, f\} + \min\{N, 2f\}$$

However, when it happens that none of the sites contains more than $f/2$ failures, then the number of signature transmission is given by:

$$\beta_0(M, N, f) = (M - \lfloor M/(2f+1) \rfloor) \min\{N, f\}$$

First consider the case where no error actually occurs. The number of signature transmission required in Algorithm GF_SIG_A, Algorithm GF_SIG_B and Algorithm GF_SIG for different sets of parameters are plotted in Figure 3. The difference between the number of signature transmission for the two algorithms is shown in Figure 4. (In these figures, $\gamma_0^A(M, N, f)$ is denoted by $Gamma(0A)$, $\gamma_0^B(M, N, f, h, F)$ is denoted by $Gamma(0A)$, $\beta_0(M, N, f)$ is denoted by $Beta(0)$.)

When comparing $\gamma_0^A(M, N, f)$ with $\beta_0(M, N, f)$, Note that the minimum value of $\beta_0(M, N, f)$ occurs at the smallest possible value of f, namely 1, in which case $\beta_0(M, N, f) = \lceil \frac{2M}{3} \rceil \min\{N, f\}$. From Figure 4(a), the savings of Algorithm GF_SIG_A on the message transmission approximately ranges from $\frac{M}{6} \min\{N, f\}$ at $f = 1$ to $\frac{M}{2} \min\{N, f\}$ at $f = \lfloor M/2 - 1 \rfloor$. It can be seen that the enhancement in performance of Algorithm GF_SIG_A and Algorithm GF_SIG_B over Algorithm GF_SIG increases with the number of errors to be detected.

5.4. *Cases with Failures*

The maximum numbers of signature transmission required for Algorithm GF_SIG_A and Algorithm GF_SIG_B under different system parameters of N, M, f, g and F are shown in Figure 5. The requirements of Algorithm GF_SIG are also shown. In the remaining figures, $\beta(M, N, f)$ is denoted by $Beta(w)$, and $\beta_0(M, N, f)$ is denoted by $Beta(0)$. $\gamma^A(M, N, f)$ and $\gamma^B(M, N, f, g, h, F)$ are denoted by Gamma(A) and Gamma(B), respectively.

The value $\gamma_0^B(M, N, f, h, F)$ for the case of no failure ($F = 0$) is a lower bound on the value of $\gamma^B(M, N, f, g, h, F)$. From the figure, it is seen that the value of $\gamma^B(M, N, f, g, h, F)$ is furthest away from $\gamma_0^B(M, N, f, h, F)$ for a relatively unreliable system, where the number of possible failing sites is almost half the number of sites (Figure 5(a)). The difference becomes smaller as the number of possible failing sites is decreased (Figure 5(c)).

5.4.1. Favourable Conditions for Algorithm GF_SIG_A. The previous performance comparisons show that both Algorithm GF_SIG_A and Algorithm GF_SIG_B have better performance than Algorithm GF_SIG. Next the choice between Algo-

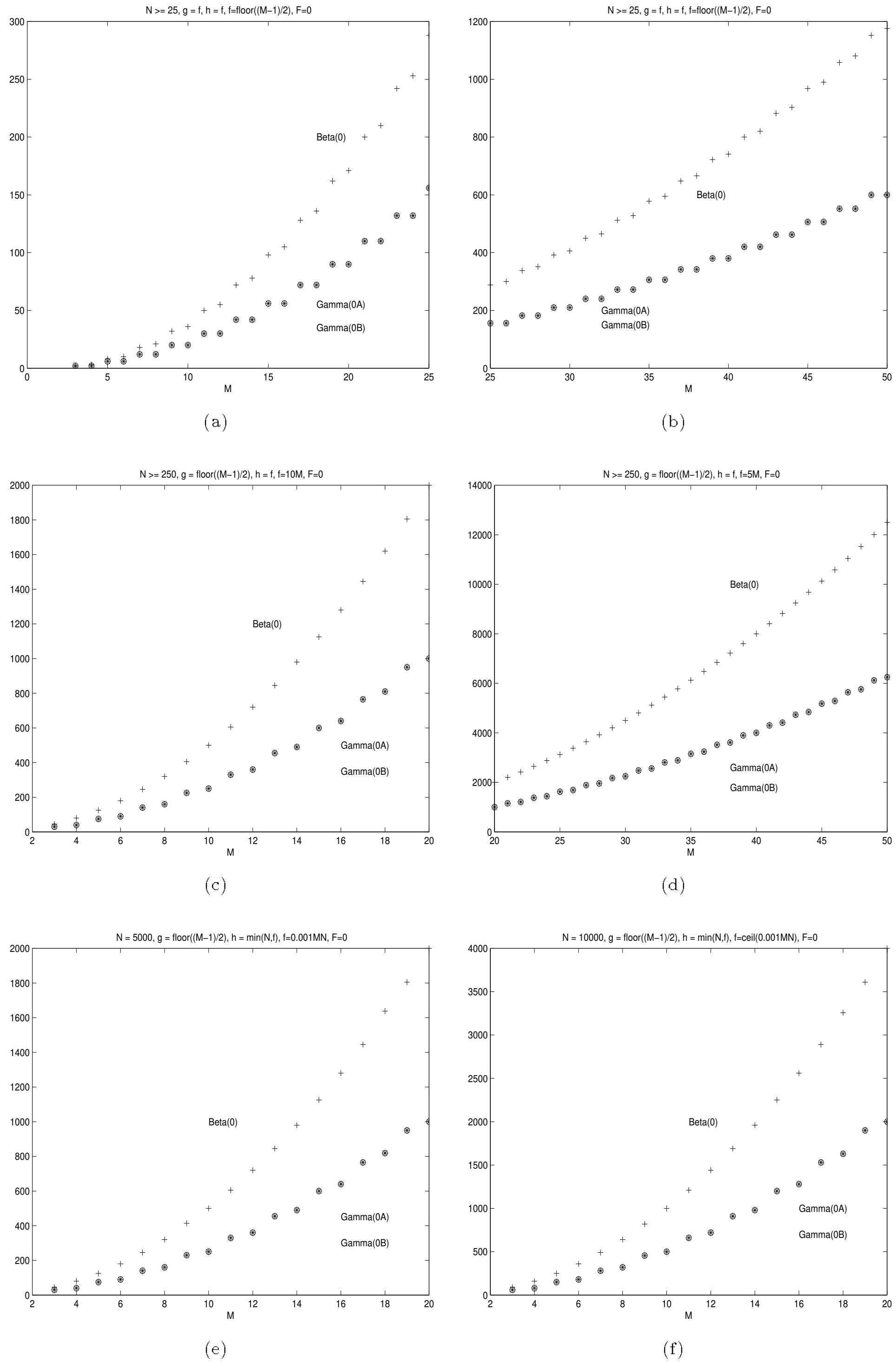
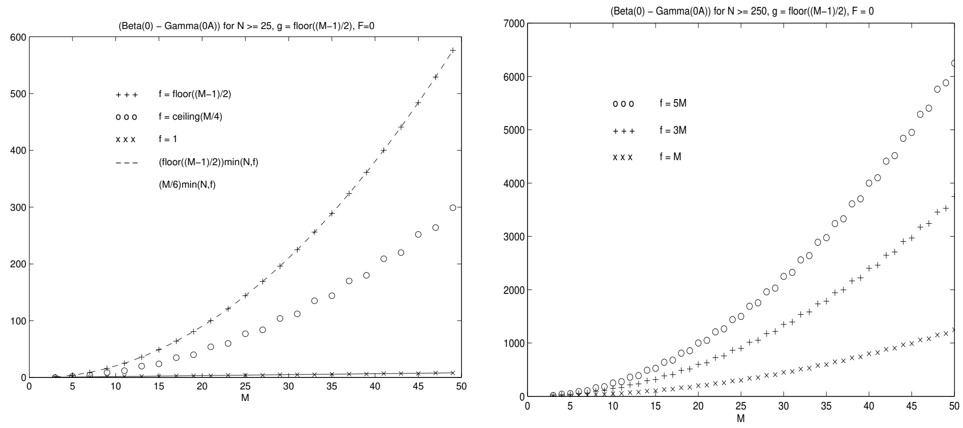


Figure 3: Signature transmission of Algorithm GF_SIG_A, Algorithm GF_SIG_B and GF_SIG under no failure



(a) (b)

Figure 4: Difference between Algorithm GF_SIG_A and GF_SIG under no failure

rithm GF_SIG_A and Algorithm GF_SIG_B is examined. Algorithm GF_SIG_A will be shown to be a good choice if the parameter of h is not considered. Figure 6 shows the performance of Algorithm GF_SIG, Algorithm GF_SIG_A, and Algorithm GF_SIG_B, without considering the parameters h and g, that is, h and g are set to their greatest possible values, respectively. The values of $\beta(M,N,f)$, $\gamma^A(M,N,f)$, $\gamma^B(M,N,f,g,h,F)$ are plotted for $F = M/5$ and $F = M/10$, for each of $f = 2M$ and $f = 0.001MN$.

From the figures, if F is small compared to f, then Algorithm GF_SIG_A performs better than Algorithm GF_SIG in that the amount of messaging is less. These diagrams show that Algorithm GF_SIG_AA is a better choice if the parameter of h is not considered.

5.4.2. Favourable Condition for Algorithm GF_SIG_B: The parameter h. Figure 7 illustrates the significance of parameter h, the maximum number of corrupted page copies at a site. If this is not known, h may be assumed to be $\min\{N,f\}$. However, it is not likely that all pages of a file are corrupted. In the figures, some reasonable values of h, such as 10% of the total number of pages, are assumed. Compared with Figure 5, the number of required signature transmission are reduced significantly. Since only Algorithm GF_SIG_B takes advantages of h, it is the best choice if a reasonable value of h is known.

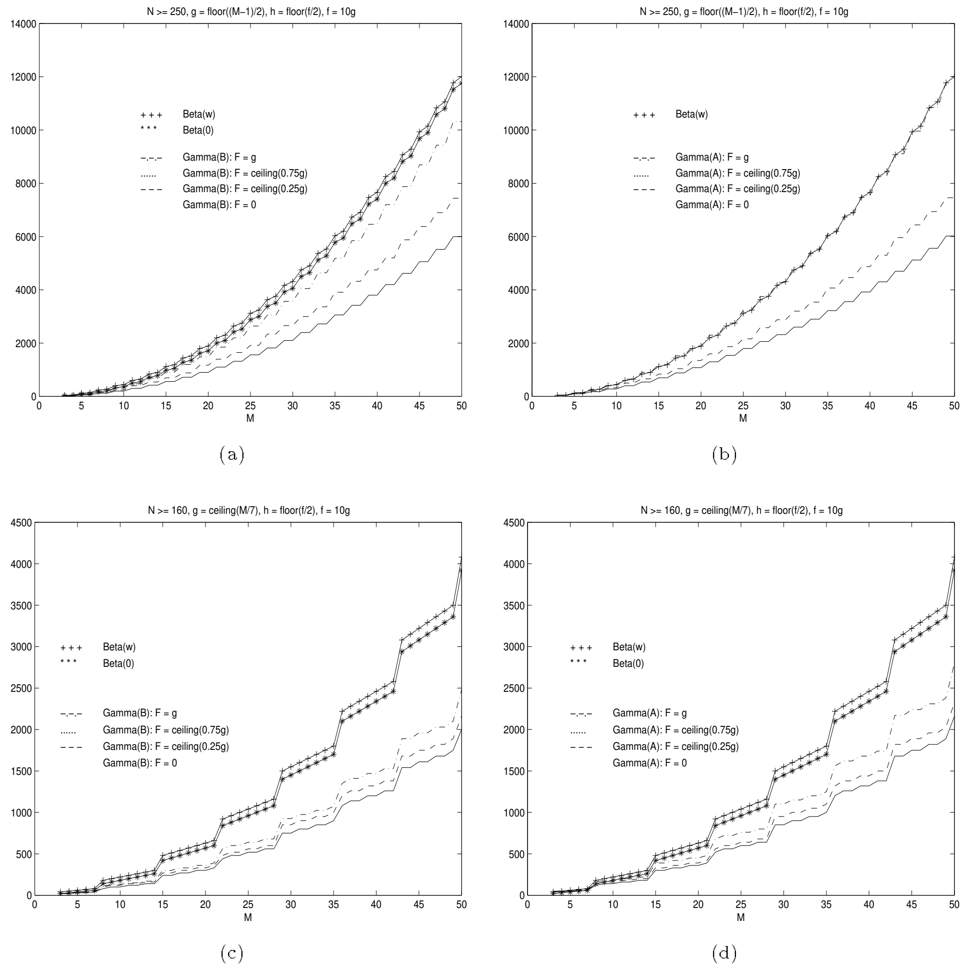


Figure 5: Number of signature transmissions for Algorithm GF_SIG_A, Algorithm GF_SIG_B and Algorithm GF_SIG with failure

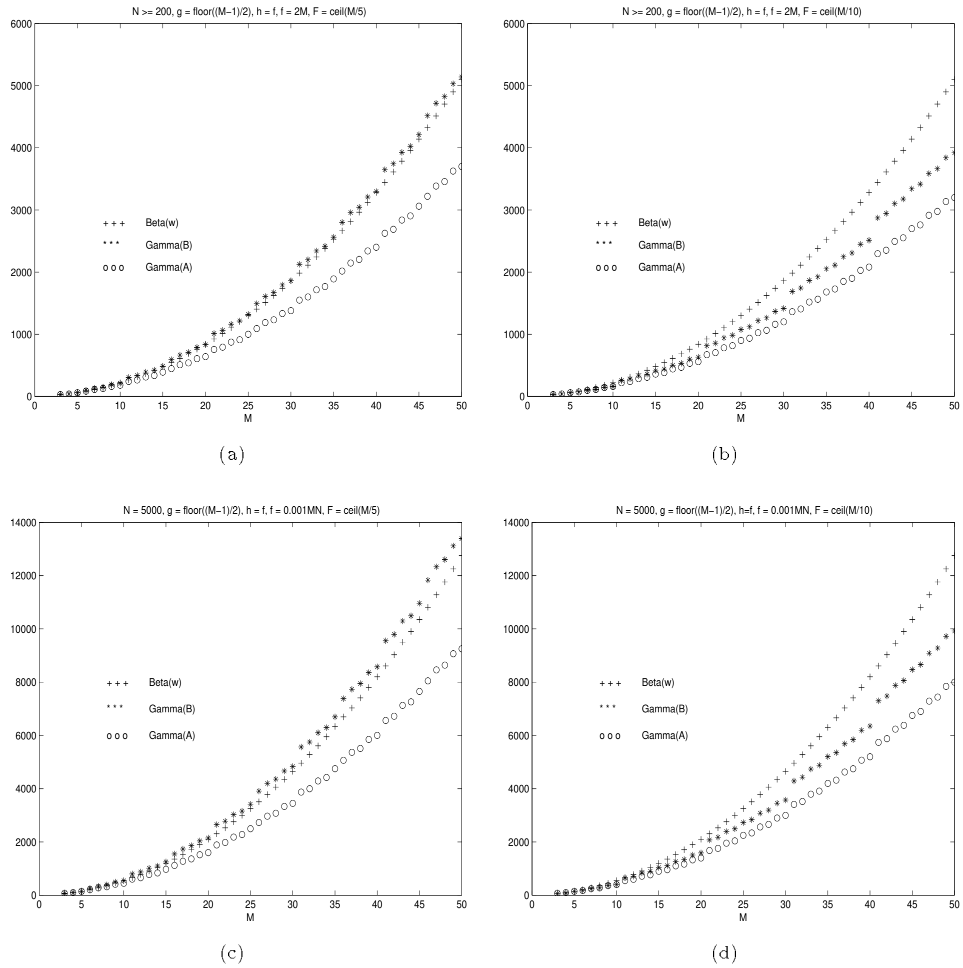
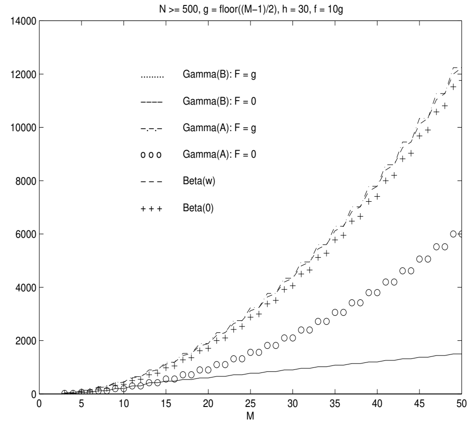
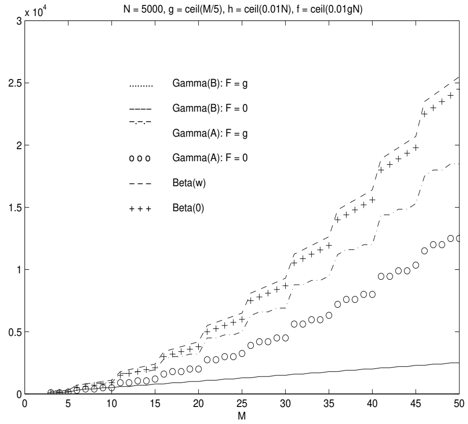


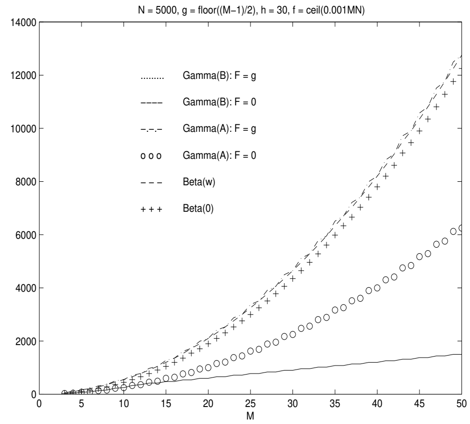
Figure 6: Signature transmission of Algorithm GF_SIG, Algorithm GF_SIG_A and Algorithm GF_SIG_B with failure



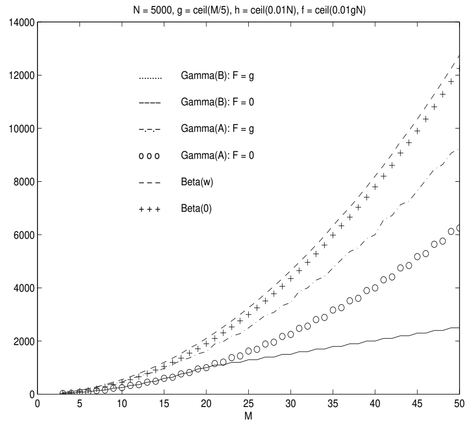
(a)



(b)



(c)



(d)

Figure 7: Effects of reasonable values of h for Algorithm GF_SIG_B

6. A Lower Bound and Optimality

It can be shown that Algorithm GF_SIG_A and Algorithm GF_SIG_B transmit the minimal number of combined signatures for the case where no failure occurs. Any failure that could have occurred must be detected. Therefore each site must compare its combined signatures with some other site in order to determine if it is correct. The minimal number of site comparisons is $\lceil M/2 \rceil$. If less number of comparison is made, then there must be at least one site which has not compared with any other site.

Reasoning similar to that in [1, 2] is applied in our proof of optimality. When 2 sites compare to discover if either one or both is corrupted, if there can be at most k failures in the 2 sites, then they must compare at least $\min\{N, k\}$ combined signatures. Assume this is not true, i.e. the number of combined signatures t compared is fewer than $\min\{N, k\}$. If the signature size is b, then t combined signatures can distinguish 2^{bt} different sets of page signatures. For a given pattern in page signatures $S(P_{\min\{N,k\}+1}), ..., S(P_N)$, there are $2^{b \min\{N,k\}}$ different patterns for the entire set of page signatures. Since t is less than $\min\{N, k\}$, there are two sets of page signatures $p = (S(P_1), ..., S(P_{\min\{N,k\}}), S(p_{\min\{N,k\}+1}), ..., S(p_N))$ and $p' = (S(p'_1), ..., S(p'_{\min\{N,k\}}), S(p_{\min\{N,k\}+1}), ..., S(p_N))$ that agree on the last $N - \min\{N, k\}$ pages but disagree in at least one page signature among the first $\min\{N, k\}$ pages, such that the combined signatures of these two sets are identical in the comparison. Hence there is no way to discover the difference if one site contains the set of page signatures p and another site contains the set of page signatures p'.

Under the system model for Algorithm GF_SIG_B, the number of possible differing copies of 2 sites is given by $\min\{N, f, 2h\}$. Under the system model for Algorithm GF_SIG_A, the number of possible differing copies of 2 sites is given by $\min\{N, f\}$. In practice, if comparison is made one by one, we may discover some corruptions in the earlier comparisons and reduce this number of possible errors for later comparisons. However, in the worse case, all the errors can occur at the site(s) in the the last comparison, so that this number cannot be reduced for any of the comparisons. Therefore, in the worst case, the number of combined signatures needed to be transmitted is given by $\lceil \frac{M}{2} \rceil \min\{N, f, 2h\}$ ($\lceil \frac{M}{2} \rceil \min\{N, f\}$). This is achieved by Algorithm GF_SIG_B and GF_SIG_A when no corruption occurs (i.e. F = 0).

LEMMA 3 *: With the assumptions in Algorithm GF_SIG_B, the minimum number of combined signature transmission in the case of no failure is given by*

$$\lceil M/2 \rceil \min\{N, f, 2h\}.$$

LEMMA 4 : *With the assumptions in Algorithm GF_SIG_A, the minimum number of combined signature transmission in the case of no failure is given by*

$$\lceil M/2 \rceil \min\{N, f\}.$$

7. Relaxing Assumption A

Assumption A states that the probability that 2 copies of the same page are corrupted and that their signatures are identical is negligible. Now it is shown how this assumption may be relaxed. Assumption A is replaced by the following assumption, where $R \geq 2$:

Assumption C: *the probability that any R corrupted copies of a page have identical signatures is negligible.*

With Assumption C, for $X < R$ corrupted copies of a page, the error patterns can be identical (the chance is not assumed to be negligible). Hence when there are X sites showing identical signatures, it cannot be concluded that there is no corruption. However, with R sites, if at least one copy is corrupted but not all copies are corrupted, the differences in the signatures of the corrupted copies as compared to those of the correct site(s) will be seen; if all R copies are corrupted, at least one copy would have a corruption pattern that is different from some other copy. Hence, when it is found that all signatures are identical in R copies, one can conclude that all R copies are correct.

It may also be desirable to relax the second assumption of Algorithm GF_SIG_A and Algorithm GF_SIG_B which states that $M > 2g$. Note that this assumption and Assumption A are actually related. With Assumption C, Phase 1 of Algorithm GF_SIG_A and Algorithm GF_SIG_B may be modified as follows: Partition the set of sites into groups of $k(k \geq R)$ sites so that one leader site in each group gets to know the necessary combined signatures of all sites in the group. Note that any R sites which contain identical signatures will be correct sites. In order to prevent such a set from appearing in a group of k sites, at least $k - R + 1$ sites must be corrupted in the group. Therefore, the assumption $M > 2g$ can be replaced by

$$M > \frac{k}{k - R + 1} g, \quad k \geq R.$$

When $k = R$, $M > Rg$. For $R = 2$, $M > \frac{k}{k-1} g$, where k is an integer ≥ 2. Using larger values of k implies that more site failures can be tolerated, but it would also increase the number of signature transmissions in cases where there is no corrupted site.

LEMMA 5 : *Given Assumption C, the value of g is bounded by*

$$M > \frac{k}{k - R + 1} g, \quad k \geq R.$$

8. Summary and Conclusion

By making the realistic assumption that different sites are unlikely to contain corrupted copies of a page of a replicated file with identical page signatures, three algorithms are proposed for locating corruptions which have advantages over previous methods. The first algorithm is based on a system model as in previous work, and achieves better performance in terms of the number of signature transmission than previous work. The second algorithm is based on a finer model in which the maximum number of corrupted pages in each site is defined, and is shown to have further enhancements in the amount of signature transmission. The third algorithm allows more tolerance in data corruption compared to previous methods. The first two algorithms are also shown to be optimal in terms of the number of signature transmission in the case of no failure.

Notes

1. However, in this case of partitions of size 3, in order for at least one group to discover a correct file copy, it is no longer required that $M > 2g$, one need only require that $M > \frac{3}{2}g$. There will be more discussion about this in Section 7.
2. Note that the analysis can be refined 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 K is taken into consideration in the analysis, better performance can be achieved. This is not considered for the sake of simplicity.

References

1. Abdel-Ghaffar, K.A.S. and El Abbadi, A. (1993). Efficient Detection of Corrupted Pages in a Replicated File *12th ACM Symposium on Principles on Distributed Computing.* Pages 219-227.
2. Abdel-Ghaffar, K.A.S. and El Abbadi, A. (1993). Comparing multiple file copies with a primary copy using minimal communication. *Technical Report TRCS 93-8, Department of Computing Science, University of California at Santa Barbara.* May 1993.
3. Abdel-Ghaffar, K.A.S. and El Abbadi, A. (1994). An optimal strategy for comparing file copies", *IEEE Transactions on Parallel and Distributed Systems.* Jan 1994, Pages 87-93.

4. Barbara, D. and Garcia-Molina, H. and B. Feijoo. Exploiting symmetries for low-cost comparison of file copies. (1988). *Proc. Int. Conf. Distributed Comput. Syst.* June 1988. Pages 471-479.
5. Blahut, R.E. (1984) Theory and Practice of Error Control Codes. Addison Wesley, Reading, MA.
6. A. Fu and S. C. Chau. (1996). Locating more corruptions in a Replicated File. *15th Symposium on Reliable Distributed Systems(SRDS).* Pages 168-177.
7. Fuchs, W. and Wu, K.L. and Abraham J.A. (1986) Low-cost comparison and diagnosis of large remotely located files. *Proc. Symp. Reliability of Distributed Software and Database Systems.* Jan 1986. Pages 67-73.
8. Metzner, J.J. (1983) A parity structure for large remotely located replicated data files. *IEEE Transactions on Computers.* Aug 1983. C-32:727-730.
9. Metzner, J.J. (1986). Reliable and efficient broadcast of files to a group of locally interconnected stations. *Proc. GLOBECOM '86.* Pages 1762-1767.
10. Metzner, J.J. (1991). Efficient replicated remote file comparison. IEEE Transactions on Computers. May 1991, C-40(5):651-660.
11. Metzner, J.J. and Kapturowski. (1990). A general decoding technique applicable to replicated file disagreement location and concatenated code decoding. *IEEE Transactions on Inform. Theory.* July, 1990. IT-36:911-917.

Journal of Supercomputing, 5, 35–34 (1997)
© 1997 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Contributing Authors

D. K. Hammer initially studied Technical Physics at the Technical University of Vienna and attained his masters degree in 1966. During the next 9 years, he was a staff member of the physics department: first as a research assistant, then as an associate professor and finally as an assistant professor. During this period, his interest shifted gradually from physics via electronics to computer science.

His last position was head of the data acquisition and process control group. In 1976 he switched to industry and became head of a system software development department at Philips Vienna. In this function he was responsible for the architecture and software development of the Philips P5000 word-processing system and the Philips P2000 personal computer.

Since 1987 he is a full professor at the Computing Science Department of the Technical University of Eindhoven, The Netherlands. His research area is the construcetion of embedded systems for process control applications, including the networking and software engineering aspects.

In addition to its academic position, Prof. Hammer is active in many committees hes strong relations with many industrial companies where he also works as an advisor.

Table of Contents: Volume 5 (1997)

Number 1

Numbers 2/3 (Special Issue on Computational Learning Theory)

Number 4

Journal of Supercomputing

An International Journal

Volume 5, No. 3, July 1997