A THEORETICAL FOUNDATION FOR THE WEIGHTED CHECKSUM SCHEME

CYNTHIA J. ANFINSON

Center for Applied Mathematics, Cornell University
Ithaca, New York 14853

RICHARD P. BRENT

Computer Sciences Laboratory, Australian National University
Canberra, ACT 2601, Australia

FRANKLIN T. LUK

School of Electrical Engineering, Cornell University
Ithaca, New York 14853

ABSTRACT

The weighted checksum scheme has been proposed as a low-cost error detection procedure for parallel matrix computations. Error correction has proved to be a much more difficult problem to solve than detection when using weighted checksums. In this paper we provide a theoretical basis for the correction problem. We show that for a distance $d+1$ weighted checksum scheme, if a maximum of $\lfloor d/2 \rfloor$ errors ensue then we can determine exactly how many errors have occurred. We further show that in this case we can correct the errors and give a procedure for doing so.

1. INTRODUCTION

Algorithm-based fault tolerance was originally proposed by Huang and Abraham [1] and later refined by Jou and Abraham [2]. Based on ideas from coding theory, an input vector or matrix is encoded with weighted checksums and the algorithm operates on the encoded input to produce encoded output. The encoded output is then checked for errors. Weighted checksums have been shown to be effective for several applications areas including matrix addition, matrix multiplication, and triangular decompositions [1], [2], [3].

In previous work [2], [4] a linear algebraic interpretation of the weighted checksum scheme has been proposed. The importance of the linear algebraic model is that it allows parallels to be drawn between algorithm-based fault tolerance and coding theory. In particular, it has been shown [4] that in a distance $d+1$ code a maximum of d errors can be detected and a maximum of $\lfloor d/2 \rfloor$ errors can be uniquely corrected. The model makes it possible to examine in detail the difficulties in choosing weight vectors such that the correction vector can be explicitly resolved. However, it has not yet been demonstrated how to determine the exact number of errors that have occurred. We show, under certain assumptions, how to decide upon the exact number of errors. Error correction has been demonstrated to be a more difficult problem to handle than detection. It has been previously shown how to correct a weighted checksum scheme for the cases of one error [2] and two errors [4]. In this paper we will give a theoretical framework which will allow us to solve the correction problem. It should be stressed that our correction procedure assumes only an upper bound of $\lfloor d/2 \rfloor$ on the number of errors.

This paper contains three major sections. First, important background material is presented to lay a foundation for the discussion that follows. Then we prove that, if a maximum of $\lfloor d/2 \rfloor$ errors ensue in a distance $d+1$ code the errors can be detected and we can also determine exactly how many have occurred. Finally, improving on previous results, we show that we can always correct the errors in the aforementioned case and a procedure is given for doing so. It is to be stressed that these results are theoretical in nature, and are not algorithmic. Furthermore, the new results use a more general set of weights than those previously considered [2], [4].

A remark about the use of notation in the sequel. A subscripted variable with a superscript is to be interpreted as a subscripted variable raised to the power given by the superscript; i.e., $w_i^j = (w_i)^j$. A subscripted variable with a superscript in parentheses such as $w_j^{(i)}$ is the j^{th} element of the vector $w^{(i)}$.

2. BACKGROUND

In the work of Huang and Abraham [1], Jou and Abraham [2], and Anfinson and Luk [4], the linear algebraic model of the weighted checksum scheme is developed. We now briefly review this important background material. We also discuss the fault model relevant to our results, and several important assumptions and their implications.

2.1 Definitions

The weighted checksum matrix H is similar to the parity check matrix in coding theory and is said to generate the code.

Definition 1: The $d \times (n+d)$ WC (weighted checksum) matrix H is given by

$$H = \begin{bmatrix} w_1^{(1)} & w_2^{(1)} & \cdots & w_n^{(1)} & -1 & 0 & \cdots & 0 \\ w_1^{(2)} & w_2^{(2)} & \cdots & w_n^{(2)} & 0 & -1 & \cdots & 0 \\ \cdot & \cdot & \cdots & \vdots & 0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ w_1^{(d)} & w_2^{(d)} & \cdots & w_n^{(d)} & 0 & 0 & \cdots & -1 \end{bmatrix}. \tag{1}$$

Definition 2: The code space C of H consists of all vectors which lie in the null space $N(H)$ of H, where $N(H) = \{ x : Hx = 0 \}$.

In order to make precise the meaning of distance in the code space we define a metric upon the domain of H. It is easily checked that the metric satisfies the properties of a distance [4].

Definition 3: The distance between two vectors v and w in the domain of H, $dist(v, w)$, equals the number of components in which v and w differ.

Definition 4: The distance of the code space C is the minimum of the distances between all possible pairs of nonzero vectors in $N(H)$; i.e., distance of C = min $\{dist(v, w) : v, w \text{ in } N(H), v \neq 0, w \neq 0\}$.

Definition 5: Let x be in $N(H)$, and $\hat{x}$ be a possibly erroneous version of x. Define the syndrome vector s by $s = H\hat{x}$, and the correction vector c by $c = \hat{x} - x$.

Note that $Hc = s$ since $Hc = H\hat{x} - Hx = H\hat{x} = s$. For simplicity we will subscript the vectors c and s as follows; $c = (c_0, c_1, \cdots, c_{n+d-1})$ and $s = (s_0, s_1, \cdots, s_{d-1})$.

We can now make precise the definitions of error detection and correction. Let α denote the total number of errors that have occurred and define γ by

$$\gamma = \left\lfloor \frac{d}{2} \right\rfloor$$

A coding scheme can *detect* α errors if the syndrome vector s is nonzero whenever $1 \le \alpha \le d$. A coding scheme can *correct* α errors if $\hat{x}$ can be corrected to x for $1 \le \alpha \le \gamma$. The next result states the sufficient conditions for error detection and correction in a distance $d+1$ code [4].

Property 1: If every set of d columns of H is linearly independent, then the distance of the code is $d+1$, a maximum of d errors can be detected, and a maximum of γ errors can be corrected.

2.2 Fault model and assumptions

Our choice of fault model has been directly influenced by our area of applications interest. Multiprocessor systems for real time digital signal processing (e.g., systolic arrays) are considered and hence our fault model assumes that a module (a processor or computational unit) makes arbitrary logical errors in the event of a fault. Assuming that the system is periodically checked for permanent and intermittent errors, we are focusing our attention on soft errors. We suppose that the soft error rate is small under normal operating conditions; this is a fairly reasonable assumption as it has been reported that the soft error rate for a large VLSI chip (1 cm^2) is 10^{-4} per hour [5].

Two major assumptions are made throughout this work. First it is assumed that no errors occur in the checksums themselves. It should be noted that for error detection this assumption is not needed, but for our correction procedure it is a very necessary condition. It is also assumed that $1 \le \alpha \le \gamma$. The following example will illustrate the problems that can occur when $\alpha > \gamma$.

Example 1: Let $n = 6$ and $d = 2$. Then $\gamma = 1$. Let H be the following matrix.

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & 0 \\ 1 & 2 & 2^2 & 2^3 & 2^4 & 2^5 & 0 & -1 \end{bmatrix}.$$

Now suppose that $\alpha > \gamma$, say $\alpha = 2$. If s is given by

$$s = \begin{bmatrix} 1 \\ 4 \end{bmatrix},$$

then any two of the first six elements of c can be solved for since we are assuming that no errors occur in the checksums. The following two values of c both satisfy $Hc = s$.

$$c = \begin{bmatrix} -2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T,$$

and

$$c = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Note that c is not unique. In fact, the second value of c is actually the solution for the correction vector when $\alpha = 1$. Thus we cannot differentiate between a case with one error and a case with two.

3. THE DETECTION PROCEDURE

From the previous section it is known that we can detect α errors when $0 \le \alpha \le d$. Assume that H, a $d \times (n+d)$ matrix, has the form

$$H = \begin{bmatrix} \psi_0^0 & \psi_1^0 & \cdots & \psi_{n-1}^0 & -1 & 0 & \cdots & 0 \\ \psi_0^1 & \psi_1^1 & \cdots & \psi_{n-1}^1 & 0 & -1 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & 0 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdot & \cdots & \cdot \\ \psi_0^{d-1} & \psi_1^{d-1} & \cdots & \psi_{n-1}^{d-1} & 0 & 0 & \cdots & -1 \end{bmatrix}. \quad (2)$$

In the notation used in equation (1) let $w^{(i)} = (\psi_0^{i-1}, \psi_1^{i-1}, \cdots, \psi_{n-1}^{i-1})^T$, for $i = 1, 2, \cdots, d$. It can be proved that for $\psi_i \ne \psi_j$, $i \ne j$, every set of d columns of H is linearly independent [4].

The syndrome vector s was defined as $s = Hc$. Since we have assumed that no errors occur in the checksums themselves we see that elements $c_n = c_{n+1} = \cdots = c_{n+d-1} = 0$. Hence, the j^{th} element of s is given by the formula

$$s_j = \sum_{k=0}^{n-1} \psi_k^j c_k, \quad \text{for } j = 0, 1, \cdots, d-1.$$

Clearly there are only α nonzero elements in the set $\{c_0, c_1, \cdots, c_{n-1}\}$ because by definition $c = \hat{x} - x$, and $\hat{x}$ differs from x by α elements. Denote the nonzero elements of c by c_{i_k}, for $k = 1, \cdots, \alpha$. Hence, s_j may be expressed as

$$s_j = \sum_{k=1}^{\alpha} \psi_{i_k}^j c_{i_k}, \quad \text{for } j = 0, 1, \cdots, d-1. \quad (3)$$

We would like to express s_j in terms of known parameters and at this time α is unknown and γ is known. So, let

$$y_k = c_{i_k} \text{ and } \xi_k = \psi_{i_k}, \text{ for } k = 1, \cdots, \alpha.$$

For $k = \alpha+1, \cdots, \gamma$ let $y_k = 0$ and select $\xi_k = \psi_j$ such that $\xi_l \ne \xi_i$ for all $i \ne l$, where $i, l = 1, \cdots, \gamma$. With these substitutions s_j becomes

$$s_j = \sum_{k=1}^{\gamma} \xi_k^j y_k, \quad \text{for } j = 0, 1, \cdots, d-1. \quad (4)$$

In order to solve the problem $Hc = s$ for c, we need to find $y = (y_1, y_2, \cdots, y_\alpha)^T$ and $\xi = (\xi_1, \xi_2, \cdots, \xi_\alpha)^T$, since the elements of ξ give the locations of the errors and the elements of y contain the correction values. Let K be the following $\gamma \times \gamma$ symmetric Hankel matrix

$$K = \begin{bmatrix} s_0 & s_1 & \cdots & s_{\gamma-1} \\ s_1 & s_2 & \cdots & s_\gamma \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ s_{\gamma-1} & s_\gamma & \cdots & s_{2\gamma-2} \end{bmatrix}. \tag{5}$$

The first theorem provides a factorization of K which is of interest in that it allows us to determine the exact value of α. A priori knowledge of α will later prove important in the correction procedure.

Theorem 1: The matrix K has the factorization $K = VYV^T$ and is of rank α, where α is the number of errors that have occurred,

$$V = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_1 & \xi_2 & \cdots & \xi_\gamma \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \xi_1^{\gamma-1} & \xi_2^{\gamma-1} & \cdots & \xi_\gamma^{\gamma-1} \end{bmatrix},$$

and

$$Y = diag\,[y_1, y_2, \cdots, y_\gamma].$$

Proof: Multiplying out VYV^T we find that the (i,j) entry is equal to;

$$\sum_{k=1}^{\gamma} y_k \xi_k^{i+j-2} = s_{i+j-2},$$

using the definition of s_j (see equation (4)). Similarly, we see that

$$(K)_{ij} = s_{i+j-2}.$$

Hence $K = VYV^T$. To prove the second part of our claim, namely that $rank(K) = \alpha$, we note that $rank(K) = rank(Y)$. This is true because rank is not affected by multiplication by the nonsingular matrices V and V^T. V and V^T are easily seen to be nonsingular since $\xi_i \neq \xi_j$ for $i \neq j$ and both are Vandermonde. Since Y is a diagonal matrix, its rank is equal to the number of nonzero diagonal elements, namely α. Thus, $rank(K) = \alpha$. □

In order to determine precisely the number of errors that have occurred we need to determine the rank of K. Note that K is a symmetric matrix; thus we can compute its eigenvalue decomposition accurately and hence its rank can be determined [6]. As we will see in the sequel, we will need to have the matrix K to be of full rank in order for the correction procedure to work effectively. The following corollary to Theorem 1 will allow us to handle the case of a rank deficient K. Define K_α as the leading $\alpha \times \alpha$ principal submatrix of K (see (5)).

$$K_\alpha = \begin{bmatrix} s_0 & s_1 & \cdots & s_{\alpha-1} \\ s_1 & s_2 & \cdots & s_\alpha \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ s_{\alpha-1} & s_\alpha & \cdots & s_{2\alpha-2} \end{bmatrix}. \tag{6}$$

Corollary 1.1 presents a simple decomposition for K_α.

Corollary 1.1: The matrix K_α can be factored as $K_\alpha = V_\alpha Y_\alpha V_\alpha^T$ and has rank α, where

$$V_\alpha = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \xi_1 & \xi_2 & \cdots & \xi_\alpha \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ \xi_1^{\alpha-1} & \xi_2^{\alpha-1} & \cdots & \xi_\alpha^{\alpha-1} \end{bmatrix},$$

and

$$Y_\alpha = diag\,[y_1, y_2, \cdots, y_\alpha].$$

Proof: The proof follows easily from Theorem 1 and equation (3). □

4. THE CORRECTION PROCEDURE

Now we will give a general outline for the correction procedure. As it will be seen, a rank deficient K will cause the procedure to encounter difficulties. Hence, throughout this section we will assume that α is known and K_α is of full rank. It is also understood that if $\alpha = \gamma$ then $K_\alpha = K$.

Define a polynomial $P(z)$ whose roots are the unknown weights ξ_i, for $i = 1, , \ldots, \alpha$. We will show that we can determine the coefficients of the polynomial by solving a linear system involving the matrix K_α.

$$P(z) = \prod_{i=1}^{\alpha}(z - \xi_i) = \sum_{i=0}^{\alpha} a_i z^i, \tag{7}$$

where the coefficients of P are given by

$$a_i = (-1)^{i+\alpha} \sum_{j_1 < \cdots < j_{\alpha-i}} \xi_{j_1} \cdots \xi_{j_{\alpha-i}}, \tag{8}$$

and $a_\alpha = 1$.

Theorem 2 and its two corollaries will give us the foundation for the correction procedure. We will find necessary the following definitions.

$$\text{Let } f = (s_\alpha, s_{\alpha+1}, \cdots, s_{2\alpha-1})^T, \text{ and } F = (K_\alpha, f). \tag{9}$$

That is, f is an α element vector and F is an $\alpha \times (\alpha+1)$ matrix formed by appending f to the matrix K_α. We will also define A_{-k} as the matrix A with the k^{th} column deleted.

Theorem 2: The coefficients of the polynomial $P(z)$ satisfy the equation

$$Fa = 0,$$

where $a = (a_0, a_1, \cdots, a_\alpha)^T$.

Proof: The j^{th} row of the equation is, for $j = 0, 1, \cdots, \alpha-1$;

$$s_j a_0 + s_{j+1} a_1 + \cdots + s_{j+\alpha} a_\alpha =$$

$$\sum_{k=1}^{\alpha} \xi_k^j y_k a_0 + \sum_{k=1}^{\alpha} \xi_k^{j+1} y_k a_1 + \cdots + \sum_{k=1}^{\alpha} \xi_k^{j+\alpha} a_\alpha =$$

$$\sum_{k=1}^{\alpha} \xi_k^j y_k (a_0 + \xi_k a_1 + \cdots + \xi_k^\alpha a_\alpha) =$$

$$\sum_{k=1}^{\alpha} \xi_k^j y_k P(\xi_k) = 0$$

since $P(\xi_k) = 0$. □

Corollary 2.1: Each coefficient a_k, for $k = 0, 1, \cdots, \alpha-1$, can be computed by the formula

$$a_k = (-1)^{k+\alpha} D_k / D_\alpha, \tag{10}$$

where $D_k = \det(F_{-k})$ and $D_\alpha = \det(K_\alpha)$.

Proof: By definition $a_\alpha = 1$ (see (8)). Hence, by Theorem 2,

$$K_\alpha a = -f. \tag{11}$$

Now, apply Cramer's Rule to get the result. Note that $D_\alpha \neq 0$ by Corollary 1.1. □

Corollary 2.2: Define the polynomial $Q(z)$ as $\sum_{i=0}^{\alpha}(-1)^i D_i z^i$. Then $Q(z)$ has roots $\xi_1, \xi_2, \cdots, \xi_\alpha$.

Proof: Applying Corollary 2.1, we see that

$$Q(z) = \sum_{i=0}^{\alpha}(-1)^i D_i z^i = \sum_{i=0}^{\alpha}(-1)^i(-1)^{\alpha+i} a_i D_\alpha z^i =$$

$$\sum_{i=0}^{\alpha}(-1)^\alpha a_i D_\alpha z^i = (-1)^\alpha D_\alpha \sum_{i=0}^{\alpha} a_i z^i = (-1)^\alpha D_\alpha P(z),$$

which has roots $\xi_1, \cdots, \xi_\alpha$. Note that $D_\alpha \neq 0$ by Corollary 1.1. □

Thus, we see that in order to find ξ we need to find the vector a of coefficients of the polynomial $P(z)$, since ξ is a vector which contains the roots of $P(z)$. So we need to solve (11) for a, and then find the roots of $P(z)$. Alternatively, we could compute the D_i's and find the roots of $Q(z)$, which will give us ξ by Corollary 2.2. Using the above facts, there are four steps that need to be accomplished for correction.

Procedure:

(1) Find $rank(K)$. This can be done by finding the eigenvalues of K (a symmetric eigenvalue problem). If $\alpha = rank(K)$ is less than γ then use the matrix K_α.

(2) Solve for the coefficients $(a_0, a_1, \cdots, a_\alpha)$ of $P(z)$. Note that this can be accomplished by solving the system (11) or by using equation (10). This step requires K_α to be of full rank.

(3) Find the roots $(\xi_1, \cdots, \xi_\alpha)$ of $P(z)$ or $Q(z)$. Note that these can be computed as the eigenvalues of a certain companion matrix. This will give us the vector ξ.

(4) Find the vector $y = (y_1, y_2, \cdots, y_\alpha)^T$ by solving the system $\Xi y = s$, where

$$\Xi = \begin{bmatrix} \xi_1^0 & \xi_2^0 & \cdots & \xi_\alpha^0 \\ \xi_1^1 & \xi_2^1 & \cdots & \xi_\alpha^1 \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \xi_1^{d-1} & \xi_2^{d-1} & \cdots & \xi_\alpha^{d-1} \end{bmatrix}.$$

Actually Ξ is a nonsingular Vandermonde matrix and there are fast techniques available to solve systems of linear equations involving a Vandermonde matrix [6]. Note that solving the system $\Xi y = s$ is equivalent to solving $Hc = s$ for the nonzero values of c.

5. CONCLUSIONS

In this paper we have proved that, given a weighted checksum matrix H which defines a distance $d+1$ code, we can determine the exact number of errors and, if the total number of errors that have occurred lies between 0 and γ, then we can correct all errors. We have also considered a more general set of weights than previously used [2], [4]. Our procedure details the mathematical analysis of the problem, and can use further improvement as a numerical algorithm. The procedure will not be prohibitively expensive to implement. Furthermore, one may use a systolic array, or some other parallel machine which can solve the eigenvalue problem rapidly [7], [8]; since $d \ll n$, the size of the eigenvalue problem we need to solve is much smaller than that of the original problem.

ACKNOWLEDGEMENTS

This work was supported in part by the Joint Services Electronics Program under contract F49620-87-C-0044 at Cornell University. Cynthia Anfinson was also supported by an U.S. Army Science and Technology Graduate Fellowship.

REFERENCES

[1] K.H. Huang and J.A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE Trans. Comput., Vol. C-33, No. 6*, pp. 518-528, June 1984.

[2] J.Y. Jou and J.A. Abraham, "Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures, *Proc. IEEE, Vol. 74, no. 5, Special Issue on Fault Tolerance in VLSI*, pp. 732-741, May 1986.

[3] F.T. Luk and H. Park, "An analysis of algorithm-based fault tolerance techniques," *J. Parallel Distri. Comput., Vol. 5*, pp. 172-184, 1988.

[4] C.J. Anfinson and F.T. Luk, "A theoretical basis for algorithm-based fault tolerance," *IEEE Trans. Comput., Vol. C-37, No. 12*, Dec. 1988, to appear.

[5] D.B. Sarrazin and M. Malek, "Fault-tolerant semiconductor memories," *IEEE Computer, Vol. 17, No. 8, Special Issue on Fault-Tolerant Computing*, pp. 49-56, August 1984.

[6] G.H. Golub and C.F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 1983.

[7] R.P. Brent and F.T. Luk, "The solution of singular-value and symmetric eigenvalue problems on multiprocessor arrays," *SIAM J. Sci. Statist. Comput., 6*, pp. 69-84, 1985.

[8] G.W. Stewart, "A Jacobi-like algorithm for computing the Schur decomposition of a non-Hermitian matrix," *SIAM J. Sci. Statist. Comput., 6*, pp. 853-864, 1985.