AC 2011-209: TESTING STRATEGY IN MULTIPROCESSOR SYSTEMS WITH CUBE CONNECTIONS

Alireza Kavianpour, DeVry University, Pomona

Dr. Alireza Kavianpour received his Ph.D. Degree from University of Southern California (USC). He is currently Senior Professor at DeVry University, Pomona, CA. Dr. Kavianpour is the author and co-author of over forty technical papers all published in IEEE Journals or referred conferences. Before joining DeVry University he was a researcher at the University of California, Irvine and consultant at Qualcom Inc. His main interests are in the areas of embedded systems and computer architecture.

TESTING STRATEGY IN MULTIPROCESSOR SYSTEMS WITH CUBE CONNECTIONS

Abstract

A college-level textbook for covering testing strategy of a multi-computer system does not exist. This paper documents different methods of testing in which the author teaches in networking and fault-tolerant computing courses. System-level testing approach in multi-computer systems in particular hypercube is the subject studied in this paper. An n-dimensional hypercube multi-computer system, or an n-cube for short, contains 2^n processors each of which is a self-contained computer with its own local memory. Each processor is assigned a unique n-bit address. Two processors are linked if and only if their addresses differ in exactly one bit position. Therefore, each processor has direct communication links to n other processors. One-step testing of hypercube which involves only one testing phase during which processors test each other is discussed. Two different kinds of one-step testing are considered: one called the precise one-step testing and studied earlier by other authors and the other called the pessimistic one-step testing and studied first by author in the context of application to hypercube. One of the two main results presented here is that the degree of testability of the n-dimensional hypercube where $n > 4$, increases from n to 2n-2 as the testing strategy changes from the precise one-step strategy to the pessimistic one-step testing strategy. The other main result is the that if the fault bound, i.e., the upper bound on the possible number of faulty processors, is kept to the same number n in both cases of precise and pessimistic testing, then the pessimistic strategy requires (n/2)+1 testing links per processor whereas the precise strategy requires n testing links per processor. A procedure for selecting (n/2+1)*n/2 (2-way) links in an n-cube for use as testing links is presented.

1. Introduction

Recently hypercube multi-computer systems have become a subject of considerable interest to the instructors teaching the networking or fault-tolerant computing courses. An n-dimensional hypercube multi-computer system, or an n-cube for short, contains 2^n processors each of which is a self-contained computer with its own local memory. Each processor is assigned a unique n-bit address. Two processors are linked if and only if their addresses differ in exactly one bit position. Therefore, each processor has direct communication links to n other processors.

Preparata et al. [16] proposed an approach in which processors test each other and the test results are collected and analyzed to determine faulty processors. This approach of mutual testing based system appears to fit well with large scale homogeneous multi-computer systems [2, 4, 5, 6, 12, 14, 17]. The approach is based on the use of an assumption or a priori knowledge on the fault bound, i.e., the upper bound on the possible number of faulty processors that might exist at any given time. Two different strategies for implementing the testing approach were discussed in [16]. One strategy is called the one-step testing or the testing without repair and under this strategy there is only one testing phase and one repair/replacement phase. Therefore, efforts are made to locate all faulty processors in the testing phase and thereafter the repair phase in which all identified faulty processors are repaired follows. The other strategy is called the sequential testing or the

testing with repair and this strategy may involve several iterations of the testing and repair phases.

The maximum number of faulty processors that may exist in a system at any given time without invalidating the testing strategy is called the degree of testability of the system under the selected testing strategy. Therefore, if the fault bound known or assumed exceeds the degree of testability under the given testing strategy, then the testing strategy cannot be applied. The degree of testability is a function of the testing connections (i.e., the set of "tester-tested connections") among the processors.

The original work in [16] was based on the repair strategy in which only those processors that were truly faulty were replaced. Therefore, the strategy may be called a precise testing strategy. Later Kavianpour [7, 9, 10, 11] proposed a strategy under which a set of r or fewer processors containing all faulty processors and possibly some processors of unknown status were identified and repaired. This strategy may be called a pessimistic testing strategy. The motivating factor here is that under the precise testing strategy, a situation where a good processor is completely surrounded by t-1 or less faulty processors, where t represents the fault bound, must not occur because the status of the isolated processor cannot be determined. Therefore, the degree of testability under the precise strategy becomes low. Under the pessimistic strategy, such an isolated processor is treated as a potentially faulty processor and replaced. An important property common to both precise and pessimistic testing strategies is that no faulty processors will remain undetected and unrepaired. While the pessimistic strategy might involve wasteful replacement of some operational processors, it has an advantage over the precise strategy in that the degree of testability becomes higher.

The pessimistic testing strategy under which up to t' processors may be replaced/repaired where t' is the fault bound, is called the (pessimistic) t'/t' testing strategy [9]. It turns out that under the t'/t' strategy at most one fault-free processor may be replaced [3, 20]. In the remainder of this paper, the term pessimistic testing refers to the pessimistic t'/t' one-step testing. A procedure for finding minimum connection patterns among a given set of processor nodes that enable pessimistic testing was developed in [15]. A procedure for identifying the processors to be repaired under the pessimistic testing strategy was developed in [20]. Also an efficient (polynomial time) technique for determining the degree of testability for a given multi-computer system under pessimistic testing was developed in [18].
It has been known for quite some time that the degree of testability of an n-cube under the precise strategy is n [1,13]. This paper presents a proof that the degree of testability of the n-cube, where n > 3, increases from n to 2n-2 as the testing strategy changes from the precise one-step strategy to the pessimistic one-step testing strategy. (Although the algorithm in [18] can be used to find the degree of testability of any individual hypercube, the closed form relation between an arbitrary-sized hypercube and its testability does not follow directly from the algorithm.) When the fault bound adopted in an n-cube is the maximum number, i.e., 2n-2, then the pessimistic testing requires the use of all inter-processor links as testing links. However, it is shown here that if the fault bound is kept to the same number n in both cases of precise and pessimistic testing, then the pessimistic strategy requires only (n/2+1) testing links per processor whereas the precise strategy requires n testing links per processor. A procedure for selecting (n/2+1)*n/2 bi-directional links in an n-cube for use as testing links is also presented.

In section 2, basic terminologies are introduced together with a graph model of a multi-computer system defined in [16]. Section 3 then presents the results on the degree of testability of the n-cube under the pessimistic one-step testing strategy. The result on selection of testing links to realize the testability of degree n is presented in section 4. Section 5 provides a conclusion of the paper.

2. A Graph Model of a Multi-Computer System and System-Level Testing

A multi-computer system is represented by a graph called the testing graph. The testing graph is a digraph G(V,E), where V is the set of nodes representing processors and E is the set of directed edges representing the testing links (i.e., the inter-processor links used as tester-tested connections) between the processors. Associated with each processor there are the tester set of processors and the tested set of processors. The outcome of a test in which processor v_i tests processor v_j is denoted by a_{ij}, and $a_{ij} = 1$ if processor vi indicates that processor v_j is faulty whereas $a_{ij} = 0$ if processor vi indicates that processor v_j is fault-free. If vi is faulty, then outcome a_{ij} is unreliable. In the case of a hypercube, each testing link is bi-directional and thus can facilitate two testing links in opposing directions. A set of test outcomes of a multi-computer system that are analyzed together to determine faulty processors is called a syndrome of the system. A syndrome analyzer that orders the processors to test others via testing links, collects the test results, and determines the set of nodes to be repaired. The connection between the syndrome analyzer and the processors of a hypercube may be either point-to-point serial links or a multi-access serial bus with the broadcast capability.

Definition 1: A multi-computer system is precisely one-step t-fault testable if given the fault bound t, all the faulty processors can be correctly identified after a testing phase.

Later Kavianpour [7, 9, 10] defined t'/t' fault testability as a part of introducing the concept of pessimistic testing.

Definition 2: A multi-computer system is pessimistically one-step t'/t' fault testable if given the fault bound t', t' or fewer processors that include all the faulty processors present and possibly some processors of unknown status can be identified for replacement after a testing phase.

Some of the basic properties of the hypercube connection are now introduced. The Hamming distance d between two processors in the n-cube is the number of bit positions where the binary representations of the addresses of the two processors differ from each other. A path in a hypercube is represented as a sequence of processors in which every two consecutive processors directly connected to each other. A path is also represented as a sequence of connected edges, each representing an inter-processor link. The number of links on a path is called the length of the path in the hypercube. By the nature of the hypercube connectivity, the length of the shortest path between two processors is the same as the Hamming distance between the two. The connectivity of a cube is the minimum number of nodes whose removal results in a disconnected cube. It was shown in [1] that an n-cube has connectivity n.

Hakimi and Amin [8] obtained the following two conditions that are sufficient to assure that a system of N processors is precisely one-step t-fault testable: 1) $N \geq 2t+1$, and 2) node

connectivity $\geq$ t. Since the n-cube has connectivity n and the number of processors is 2^n, satisfies the condition, $N \geq 2n+1$, for $n \geq 3$, the n-cube is precisely one-step n-fault testable [1, 14].

3. Pessimistic One-Step Testing of Hypercube

3.1 Basic properties of the pessimistic testing strategy

The main result presented in this section (about the degree of testability of the n-cube under the pessimistic testing strategy) is built upon some properties discovered earlier. These properties are summarized below.

Property 1: A multi-computer system is pessimistically one-step t'/t' fault diagnosable if and only if the number of tester for any 2p processors is greater or equal to (t'-p+1) for $1 \leq p \leq t'$.

Therefore, the greatest t' that satisfies the condition is the degree of testability of the system under the pessimistic testing strategy.

3.2 The degree of testability of the n-cube under the pessimistic testing strategy

An important property regarding the connectivity of the n-cube that can be utilized in determining the degree of testability is the following:

Property 2: In the n-cube where, any pair of processors can be tested by at least 2n-2 other processors. If the Hamming distance between two processors is either 1 or 2, then the number of other processors that can test the two processors is exactly 2n-2.

Based on this property a faulty n-cube with the fault bound of 2n-2, all faulty processors can be removed by replacing at most 2n-2 processors.

Property 3: The degree of testability of the n-cube under the pessimistic one-step t'/t' fault testing strategy, where $n > 3$, is 2n-2.

Experiment 1: The following experiment simulated in the networking course. In a 4-cube, each of 2^4 processors can test four immediate neighbors and vice versa. According to Property 3, a 4-cube is pessimistically one-step 6/6 fault diagnosable. Therefore, if the fault bound is 6, we can repair a faulty 4-cube in one step by replacing at most 6 processors. In the 4-cube assume five processors 0, 2, 3, 5, and 9 are faulty. An analysis of the syndrome indicates that the five processors 0, 2, 3, 5, and 9 are definitely faulty and the status of processor 1 is unknown. Therefore, the system can be repaired by replacing six processors 0, 1, 2, 3, 5, and 9. Good processor 1 is replaced because it can be tested only by four other faulty processors and thus no reliable information about its status can be made available. Note that with a 4-cube the precise one-step testing strategy can be used only when the fault bound is four or less.

4. Pessimistic one-step testing of the n-cube with reduced testing connections

In the preceding section it was shown that the n-cube could be tested by use of the pessimistic one-step testing strategy with the fault bound set to as high as 2n-2. When the fault bound

adopted is the maximum number, i.e., 2n-2, then the pessimistic testing requires the use of all inter-processor links as testing connections. That is, the number of testing connections used per processor is n. Similarly, when the precise one-step testing strategy is used and when the fault bound adopted is the maximum number allowed under the strategy, i.e., n, the precise testing requires the use of all inter-processor links as testing connections. However, if the fault bound is kept to the same number n in both cases of precise and pessimistic testing, then the pessimistic strategy requires significantly fewer testing connections than the precise strategy does. In fact, the number of testing connections required under the pessimistic strategy is (n/2+1) connections per processor.

In order to prove this, a method for constructing proper testing connection patterns will be given in the following. The key issue here is how to remove (n/2-1) links per processor from an n-cube such that the remaining links can be used to facilitate n/2+1 testing connections per processor which enable the pessimistic one-step testing of the n-cube under the fault bound of n.

Procedure 1: Consider processor pi in an n-cube where $n > 3$ and $0 \leq i \leq 2^n-1$ and the binary address vector. Let r represent n/2 for the sake of convenience.

Case 1) $i < 2^{n-1}$: Remove the link from processor pi to processor pj if their address bits (i.e., binary address vector of pi and pj) differ only in one of the (r-1) least significant bit positions.

Case 2) $i \geq 2^{n-1}$: Remove the link from processor pi to processor pj if their address bits differ in one of the following bit positions: (n-r+1)-th, or (n-r+2)-th,..., or (n-1)-th bit positions.

Experiment 2: The following experiment simulated in the networking course. Consider a 4-cube. According to Case 1 of Procedure 1, links between pairs of processor (0, 1), (2, 3), (4, 5), and (6, 7) will not be used in testing since their address bits differ in the 1st bit position. Also according to Case 2 of Procedure 1, links between pairs of processors (8, 12), (9,13), (10, 14), and (11, 15) will not be used in testing since their address bits differ in the 3rd position.

Property 4: An incomplete n-cube produced by Procedure 1, has connectivity n/2+1.

Property 5: In an incomplete n-cube produced by Procedure 1, any pair of processors can be tested by at least n other processors. If the Hamming distance between two processors is either 1 or 2, then the number of processors that can test the two processors is exactly n or n+1.

Property 6: An n-cube with the fault bound of n, can be tested by the use of pessimistic one-step testing with n/2+1 testing connections per processor.

Experiment 3: A 4-cube in which n/2+1 = 3 inter-processor links emanating from each processor are used as testing connections. This 4-cube is obviously not precisely one-step 4 fault diagnosable, but it is pessimistically one-step 4/4 fault diagnosable. Through an exhaustive case analysis one can verify that the condition in the Property 1 is satisfied. For example, faulty processors 0, 1, 2, and 3 can be tested under the pessimistic one-step strategy.

A practical implication of Property 6 is that even if some links in the n-cube are broken, the one-step pessimistic testing of the 2^n processors is possible as long as the fault bound does not exceed n.

5. Conclusion

System-level testing approach in multi-computer systems in particular hypercube has been the main subject of discussion in this paper. The regular structure of a hypercube makes teaching of testing strategy easier. Through our findings, the degree of testability of the n-cube increases as the testing strategy changes from the precise one-step strategy to the pessimistic one-step strategy. The pessimistic one-step testing appears to be a highly attractive strategy for use in hypercube systems, considering the high degree of testability that it provides, the relatively small number of tester-tested connections that it uses, and the uselessness of a processor of unknown status surrounded completely by faulty processors in typical application environments. Many important questions such as the testability of different networks such as mesh, pyramid, and star with some broken links remain unanswered.

Teaching different testing strategies in a networking or fault-tolerant computing course is a relatively new subject. A college-level textbook covering testing strategy of a multi-computer system does not exist. This paper documents different methods of testing in which the author teaches in a networking and fault-tolerant computing courses. These results could be supplemented with the textbooks used in these courses.

6. References

1. J. R. Armstrong and F. G. Gray, "Fault Diagnosis in a Boolean n Cube Array of Microprocessors," IEEE Trans. on Comput., vol. C-30, pp. 587-590, Aug. 1981.
2. P. Banerjee et al., "An Evaluation of System-Level Fault Tolerance on the Intel Hypercube Multiprocessor," Proc. 18th Int'l Symp. on Fault-Tolerant Computing, pp. 362-367, 1988.
3. K. Y. Chwa and S. L. Hakimi, "On Fault Identification in Diagnosable Systems," IEEE Trans. Comput., vol. C-30, pp. 414-422, June 1981
4. A. T. Dahbura and G. M. Masson, "An O(n2.5) Fault Identification Algorithm for Diagnosable Systems," IEEE Trans. Comput., vol C-33, pp. 486-492, June l984.
5. A. T. Dahbura, "System-Level Diagnosis: A Perspective for the Third Decade" Tech. Rept. AT&T Bell Labs., 1987.
6. E. Dilger and E. Ammann, "System Level Self Diagnosis in n cube Connected Multiprocessor Networks," Proc. 14th Int'l. Symp. on Fault-tolerant Computing, pp. 184-189, 1984.
7. A. D. Friedman, "A New Measure of Digital System Diagnosis, Proc.5th Int'l. Symp. on Fault-Tolerant Computing, pp. 167-170, 1975.
8. S. L. Hakimi and A. T. Amin, "Characterization of Connection Assignment of Diagnosable Systems," IEEE Trans. Comput., vol. c-23, pp. 86-88, Jan. 1974.
9. A. Kavianpour and K.H. Kim, "A Comparative Evaluation of Four Basic System-Level Diagnosis Strategy for Hypercubes" IEEE Transactions on Reliability, vol. 41, no. 1, pp. 26-37, March 1992
10.] A. Kavianpour and A. D. Friedman, "Efficient Design of Easily Diagnosable Systems, " Proc. 3rd USA-JAPAN Computer Conf., pp. 251-257, Oct. 1978.
11. A. Kavianpour and A. D. Friedman, "Trade-Offs in System Level Diagnosis of Multiprocessor Systems," Proc. AFIPS National Computer Conference, pp. 173-181, July 1984.

12. C. Kime, "System Diagnosis," in Fault-Tolerant Computing: Theory and Techniques, D.K. Pradhan, Editor, Englewood Cliffs, NJ: Prentice-Hall, 1985.
13. J. Kuhl and S. Reddy, "Distributed Fault-tolerance for Large Multi-Processor Systems," Proc. 7th Int'l. Symp. on Computer Architecture, pp. 23-30, 1980.
14. J. Kuhl and S. Reddy, "Fault Diagnosis in Fully Distributed Systems," Proc. 11th Int'l Symp. on Fault-Tolerant Computing, pp. 100-105, 1981.
15. N. Maxemchuk and A. Dahbura, "Optimal Diagnosable System Design Using Full Difference Triangles" IEEE Trans. on Computers, vol. C-35, pp. 837-839, Sept. l986.
16. F. P. Preparata, G. Metze and R. T. Chien, "On the Connection Assignment Problem of diagnosable Systems," IEEE Trans. Elec. Comput., vol. EC-16, pp. 848-854, Dec. 1967.
17. A. K. Somani, V. k. Agarwal, and D. Avis, "A Generalized Theory for System Level Diagnosis," IEEE Trans. on Computers, vol. C-36, pp. 538-546, May 1987.
18. G. Sullivan, "A Polynomial Time Algorithm for Fault Diagnosability" IEEE Symp. on Found. of Comp. Science, pp. 148-156, 1984.
19. H. Whitney, "Congruent Graphs and the Connectivity of Graphs," Amer. J. Math., vol 54, pp. 150-168, 1932.
20. C. L. Yang , G. M. Masson, and R. Leonetti, "On Fault Identification and Isolation in ti/ti-Diagnosable Systems," IEEE Trans. on Computers, vol. C-35, pp. 639-644, July l986.