

OPTIMAL SUBCUBE ALLOCATION IN A CIRCUIT-SWITCHED FAULTY HYPERCUBE

BABACK A. IZADI

*Department of Electronic Engineering Technology,
De Vry Institute of Technology, Columbus, OH 43209, U.S.A*

and

FÜSUN ÖZGÜNER

*Department of Electrical Engineering,
The Ohio State University, Columbus, OH 43210, U.S.A*

ABSTRACT

In this paper, we present a scheme where a $(d-1)$ -dimensional subcube is allocated in a faulty d -dimensional circuit-switched hypercube in the presence of up to $2^{(d-1)}$ faulty nodes. The scheme is then extended to allocate a $(d-1)$ -dimensional subcube in the presence of a combination of faulty nodes and faulty links. Theoretical proofs and simulation results are presented to analyze the performance of the scheme.

Keywords: Fault-tolerance, hypercube, subcube allocation, reconfiguration

1. Introduction

Multiprocessors based on d -dimensional hypercubes have been widely used for a range of applications. Research efforts have been undertaken to keep the multiprocessor functional in the presence of faulty components. One of the approaches is finding the maximum dimensional fault-free subcube of a faulty hypercube^{1,2,3}. However, two faulty nodes in antipodal positions can destroy every fault-free $(d-1)$ -dimensional subcube and thus degrade the performance of the hypercube by a factor of 4. To overcome this, Chang and Bhuyan⁴ have proposed a scheme that utilizes the properties of circuit-switched communication to maintain a $(d-1)$ -dimensional subcube in the presence of up to $\lceil \frac{d}{2} \rceil$ faulty nodes. In this paper, we present a scheme for circuit-switched hypercubes where a $(d-1)$ -dimensional subcube is allocated in a faulty d -dimensional hypercube in the presence of up to $2^{(d-1)}$ faulty nodes and extend the scheme so that a functional $(d-1)$ -subcube is maintained in the presence of a combination of faulty nodes and faulty links.

2. Construction of a Functional $(d-1)$ -Dimensional Subcube

We assume that faulty nodes retain their ability to communicate. If the distribution of faulty nodes is such that every fault-free $(d-1)$ -dimensional subcube is destroyed, we

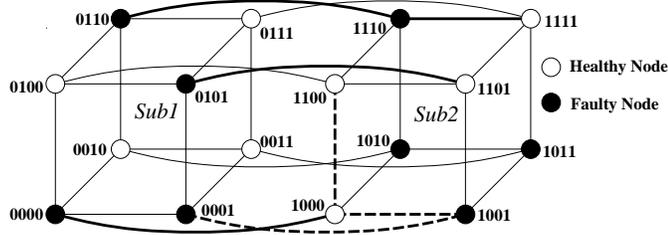


Figure 1: Construction of a 3-cube in a faulty 4-cube

use the following procedure to construct a functional $(d - 1)$ -dimensional subcube. We partition the hypercube into two faulty $(d - 1)$ -dimensional subcubes along a dimension j ($0 \leq j \leq d - 1$) and label the subcube with the fewer number of faulty nodes *Sub1* and the other subcube *Sub2*; *Sub1* and *Sub2* form a $2^{(d-1)}$ matching along dimension j . In our scheme, *Sub1* becomes a functional $(d - 1)$ -dimensional subcube by replacing each of its faulty nodes with a healthy node in *Sub2* via edge-disjoint paths along dimension j and other possible edges in *Sub2*. Fig. 1 illustrates a case in a 4-dimensional hypercube. Setting $j = 3$, the subcube 0XXX is maintained by establishing edge-disjoint paths between the faulty nodes 0000, 0001, 0101 and 0110 in *Sub1* and the healthy nodes 1000, 1100, 1101 and 1111 respectively in *Sub2*. Each of the edge-disjoint paths then becomes an extension of the communication module of the faulty node in *Sub1* and the healthy node in *Sub2* functionally replaces the faulty one in *Sub1*.

Given a faulty node $\alpha \in \text{Sub1}$, let's denote α 's neighbor along dimension j as $\beta \in \text{Sub2}$. If β is a healthy node, it can replace α . If β is faulty, then a path through β to a healthy node $\gamma \in \text{Sub2}$, that has not been assigned as a replacement, must be established. Since the communication modules of faulty nodes are assumed to be healthy, this path can go through faulty nodes as well as other assigned healthy nodes. The path $0001 \rightarrow 1001 \rightarrow 1000 \rightarrow 1100$ in Fig. 1 is an example of such a path. Note that 1000 which is part of this path also replaces faulty node 0000 via path $0000 \rightarrow 1000$. However, the two paths use disjoint edges.

In the discussion and the algorithm that follow, the set of nodes in *Sub2* are called *source nodes* (S_S) if both the nodes and their neighbors in *Sub1* are faulty. All other faulty nodes in *Sub2* and the assigned healthy nodes are referred to as the set of *used nodes* (S_U). Nodes in S_U can be intermediate nodes of a path. Finally, non-allocated healthy nodes in *Sub2* constitute the set of *target nodes* (S_T). For example, in Fig. 1, $S_S = \{1001, 1110\}$, $S_T = \{1100, 1111\}$ and $S_U = \{1000, 1010, 1011, 1101\}$. In the algorithm to be presented, nodes are assigned to these sets dynamically, as paths are established.

We next show that a functional $(d - 1)$ -subcube can always be found in a hypercube with up to $2^{(d-1)}$ faulty nodes. If every faulty node in *Sub1* is matched with a healthy node in *Sub2* along dimension j , the reconfiguration can be accomplished by simply assigning the matched healthy nodes to the faulty nodes. In the worst case, each faulty node in *Sub1* is matched with a faulty node in *Sub2* along dimension j and *Sub1* and *Sub2* each contain $2^{(d-2)}$ faulty nodes. Therefore, $2^{(d-2)}$ edge-disjoint paths from the source nodes to the target nodes in *Sub2* have to be constructed. All other cases require fewer number of edge-disjoint paths in *Sub2*. We will prove for the worst case; other cases can similarly be

proven⁵ using Menger's theorem⁶.

Lemma 1 In a d -dimensional hypercube, let a set of nodes be assigned to P and the rest be grouped under Q . The minimum number of links that connects P to Q is $\min(|P|, |Q|)$.^{7,5}

Theorem 1 In a d -dimensional hypercube, let half of the nodes be labeled source nodes and the remaining nodes be called target nodes. Within such a hypercube, there exists $2^{(d-1)}$ edge-disjoint paths to connect each source to a distinct target node.

Proof. Given a d -dimensional hypercube graph $G(V, E)$, let's partition V into two subsets P and $Q = V - P$ such that $P = S_S$ and $Q = S_T$ ($|P| = |Q| = 2^{(d-1)}$). Moreover, let's construct a new graph G' by adding two nodes s and t to G such that they be connected to every node in the set P and Q via a single edge respectively (Fig. 2). The number of edge-disjoint paths between s and t in G' , according to Menger's theorem⁶, is equal to the mincut of G' . The theorem is proven by showing that there always exists an (s, t) mincut in G' whose cutsize is greater than or equal to $2^{(d-1)}$.

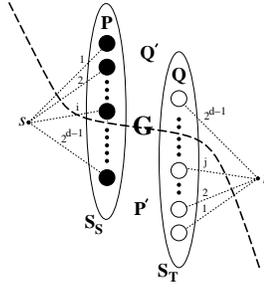


Figure 2: An (s, t) cut in G'

An (s, t) mincut in G' may exist at s , t , G , or some combination of them. By construction, the cutsize at s and t is equal to $2^{(d-1)}$. Consider a general cut in G' as depicted in Fig. 2, crossing i of the edges connecting s to the nodes within S_S , k of the edges of G , and j of the edges connecting the nodes of S_T to t ($0 \leq i, j \leq 2^{(d-1)}$). Only the case where $i + j < 2^{(d-1)}$ needs to be investigated since the other meets the minimum cutsize on its own. The cut in Fig. 2 splits the nodes into two sets P' and Q' . The number of source nodes in the set Q' is i . The number of target nodes in the same set is $2^{(d-1)} - j$. Hence, the total number of source and target nodes in Q' is $i + 2^{(d-1)} - j$. Following the same reasoning, the total number of source and target nodes in P' is $j + 2^{(d-1)} - i$. Either $|P'|$ or $|Q'|$ is less than or equal to $2^{(d-1)}$. Without loss of generality, let it be P' . From Lemma 1, the minimum number of links connecting P' and Q' is $k = \min(|P'|, |Q'|)$. Thus, the above cut must cross at least $k = j + 2^{(d-1)} - i$ links in G . The size of the (P', Q') cut is then given by $|(P', Q')| \geq i + j + j + 2^{(d-1)} - i$ or $|(P', Q')| \geq 2j + 2^{(d-1)} \geq 2^{(d-1)}$. A similar inequality results if $|Q'| < 2^{(d-1)}$, $|(P', Q')| \geq 2i + 2^{(d-1)} \geq 2^{(d-1)}$.

From the above inequalities it follows that there always exists $2^{(d-1)}$ edge-disjoint paths from s to t . Since by construction, there always exists $2^{(d-1)}$ edges from s to $2^{(d-1)}$ source nodes and $2^{(d-1)}$ edges from t to $2^{(d-1)}$ target nodes, each of the $2^{(d-1)}$ paths must connect a source node to a target node \square .

An optimal reconfiguration algorithm, to establish edge-disjoint paths between the

nodes in S_S and the nodes in S_T within $Sub2$, can be developed by utilizing the maxflow / mincut algorithm ⁶. To apply the maxflow/mincut algorithm, a digraph representation of Fig. 2 (digraph G') needs to be constructed. To avoid construction of digraph G' , we implemented a near optimal reconfiguration algorithm. The algorithm is near optimal since there can be cases where the reconfiguration fails even though Menger's theorem holds. The algorithm utilizes Lee's path-finding algorithm⁸ to find a set of candidate target nodes. It begins by constructing a breadth-first search of minimum depth in $Sub2$ from each node in S_S . If a target node is found, a path is formed between the source and the target node. The algorithm guarantees that a path to a target node will be found, if there exists one, and the path will be the shortest possible ⁸. Once a path is formed, the algorithm removes the links associated with that path from $Sub2$. It also marks both the source node and the target node as used nodes and assigns them to S_U . The process is repeated on the new resultant structure for a higher depth i . The reconfiguration is completed if S_S becomes an empty set. Reconfiguration fails if distance i becomes greater than $2^{(d-1)} - 1$ which is the longest acyclic path in $Sub2$.

3. Simulation Results

We implemented the near optimal algorithm for various dimensions of the hypercube (up to $d = 10$). 10000 simulation runs were performed for randomly placed $2^{(d-1)}$ faulty nodes. Our simulations resulted in 100% reconfiguration. To find a fault-free $(d - 1)$ -

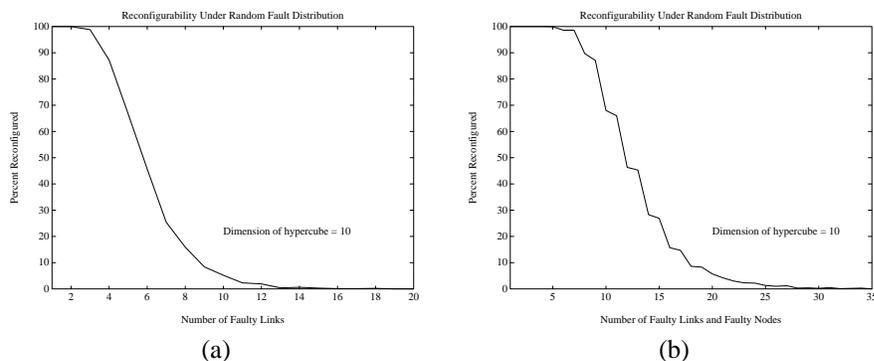


Figure 3: Percentage $(d - 1)$ -subcubes in presence of faulty (a) links (b) links and nodes

dimensional subcube in the presence of a combination of faulty nodes and faulty links, our algorithm first checks whether a $(d - 1)$ -dimensional subcube without any faulty link exists. If there exists one, it is labeled $Sub1$. Next, Lee's path-finding algorithm is used to find edge-disjoint paths from the faulty nodes in $Sub1$ to the healthy nodes in $Sub2$ via the healthy edges from $Sub1$ to $Sub2$ and the healthy edges within $Sub2$. The simulation results for a hypercube of dimension 10 under randomly placed faulty links and combination of faulty nodes and faulty links are given in Fig. 3. In our simulation we have assumed that the probability of having a node failure is the same as having a link failure. From Fig. 3, it follows that a functional $(d - 1)$ -dimensional subcube can exist provided the number of faulty links is relatively small.

4. References

1. B. Becker and H. U. Simon, "How robust is the n-cube?," *Proc. 27th Annu. Symp. Foundations Comput. Sci.*, pp. 283–291, October 1986.
2. F. Özg"uner and C. Aykanat, "A reconfiguration algorithm for fault tolerance in a hypercube multiprocessor," *Information Processing Letters*, vol. 29, pp. 247–254, November 1988.
3. N. Graham, F. Harary, M. Livingston, and Q. Stout, "Subcube fault-tolerance in hypercubes," *Information and Computation*, vol. 102, no. 2, pp. 280–314, 1993.
4. Y. Chang and L. N. Bhuyan, "Fault-tolerant subcube allocation in hypercubes," *Proceedings of the IEEE International Conference on Parallel Processing*, pp. 1132–1136, 1993.
5. B. Izadi, "Design of fault-tolerant distributed memory multiprocessors," *Ph.D. thesis, the Ohio State University*, 1995.
6. C. J. Colbourn, *The Combinatorics of Network Reliability*. Oxford University Press, 1987.
7. F. Chung, Z. Füredi, R. Graham, and P. Seymour, "On induced subgraphs of the cube," *Journal of Combinatorial Theory*, vol. 49, pp. 180–187, 1988.
8. C. Y. Lee, "An algorithm for path connection and its applications," *IRE Transactions on Electronic Computers*, vol. ec-10, pp. 346–365, 1961.