Design of a Circuit-Switched Highly Fault-Tolerant k-ary n-cube

Baback A. Izadi
Elect. Eng. Tech. Dept.
DeVry Institute of Technology
Columbus, Ohio 43209 U.S.A.
bai@devrycols.edu

Füsun Özgüner
Dept. of Elect. Eng.
The Ohio State University
Columbus, Ohio 43210 U.S.A.
ozguner@ee.eng.ohio-state.edu

Abstract

In this paper, we present a strongly fault-tolerant design for the k-ary n-cube multiprocessor and examine its reconfigurability. Our design augments the k-ary n-cube with $(\frac{k}{j})^n$ spare nodes; each set of j^n regular nodes is connected to a spare node and the spare nodes are interconnected as a $(\frac{k}{j})$-ary n-cube. Our approach utilizes the circuit-switched capabilities of the communication modules of the spare nodes to tolerate a large number of faulty nodes and faulty links without any performance degradation. Both theoretical and simulation results are presented.

1 Introduction

As the size of the k-ary n-cube multicomputer grows, due to its complexity, the probability of node and/or link failures become high. Therefore, it is crucial that such systems be able to withstand a large number of faults. To sustain the same level of performance, some researchers have investigated hardware schemes for the k-ary n-cube where spare nodes and/or spare links are used to replace the faulty ones. In the literature, a hardware scheme that retains the same service level, as well as keeping the same system topology after the occurrence of faults, is referred to as a *strongly fault-tolerant* system. Two classes of hardware schemes have been proposed in the literature. Some researchers have examined local reconfiguration techniques where a spare node can only replace a faulty node within a given subset [7, 1]. A common drawback of these approaches is low utilization of spare nodes. Moreover, the schemes do not tolerate any faulty link and generally are not strongly fault tolerant. The second class of approaches uses a global reconfiguration scheme and is based on creating a supergraph of the target topology [2, 4, 3]. The schemes are strongly fault tolerant. However, they are mostly node-minimal and suffer from large node degrees.

In this paper, we propose a global reconfiguration scheme that utilizes circuit-switched communication to make the k-ary n-cube strongly fault tolerant. Both theoretical and simulation results are presented. Our theoretical result indicates that the enhanced cluster k-ary n-cube tolerates $2n + 1$ faulty nodes regardless of the fault distribution; proofs of our theorems are omitted due to space limitation [5]. Our simulation results, based on random distribution of up to $(\frac{k}{j})^n$ faulty nodes, have yielded 100% fault coverage.

The following notations are used throughout the paper. Each node of a k-ary n-cube is identified by *n-tuple* $(a_{n-1} \cdots a_i \cdots a_0)$, where a_i is a radix k digit and represents the node's position in the i-th dimension. Each node is connected along the dimension i to the neighboring nodes $(a_{n-1} \cdots a_{(i \pm 1 mod k)} \cdots a_0)$. Each spare node, in addition to n digits, is labeled with a prefix S, *i.e.*$Sa_{n-1} \cdots a_i \cdots a_0$. The link connecting any two nodes P and Q is represented by $P \rightarrow Q$. A cluster whose local spare is labeled $Sa_{n-1} \cdots a_i \cdots a_0$ is called *cluster* $a_{n-1} \cdots a_i \cdots a_0$.

2 Overview of the Approach

An enhanced cluster k-ary n-cube (ECKN) is constructed by assigning one spare node to each group of j^n regular nodes, called a cluster; each spare node is connected to every regular node of its cluster via an intra-cluster spare link. Hence, there exist $\frac{k^n}{j^n}$ spare nodes. Furthermore, the spare nodes of neighboring clusters are interconnected using inter-cluster spare links; two clusters are declared neighbors if there exists at least one regular node in each with a direct link between them. Therefore, the topology that interconnects the spare nodes (spare network) is the $\frac{k}{j}$-ary n-cube ($j \neq \frac{k}{2}$; see [5] for $j = \frac{k}{2}$). Figure 1 depicts an enhanced cluster 6-ary 2-cube where $j = 2$. Note that the spare network is a 3-ary 2-cube. The resultant structure consists of k^n regular nodes and $(\frac{k}{j})^n$ spare nodes. The degree of each regular node and spare node is $2n + 1$ and $2n + j^n$ respectively.

We assume that faulty nodes retain their ability to communicate. This assumption may be avoided by duplicating the communication module in each node. In the k-ary n-cube with circuit-switched communication modules, the cost of communication is nearly constant between any two

Proceedings of the 1997 International Conference on Parallel Processing (ICPP'97)
0-8186-8108-X 97 $10.00

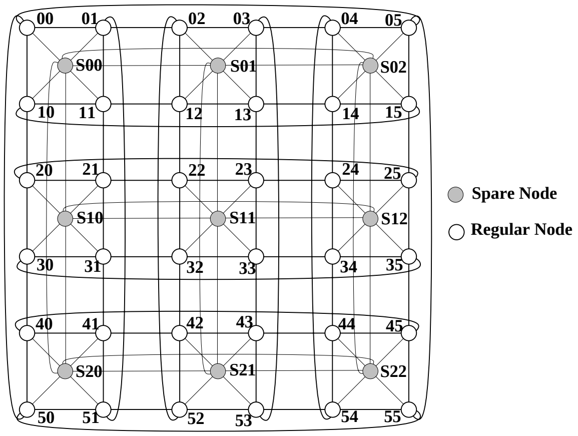

Figure 1: An enhanced cluster 6-ary 2-cube with $j = 2$

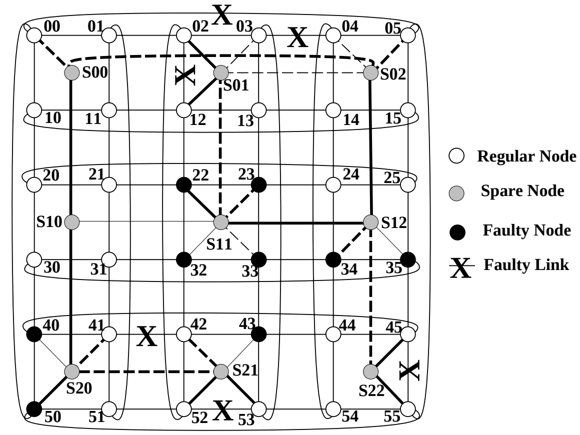

Figure 2: Reconfiguration of an ECKN.

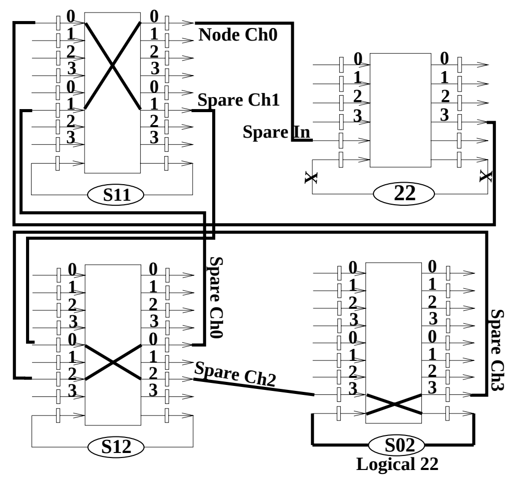

Figure 3: Tolerating faulty node 22 with spare node S02

given nodes. The regular node router consists of $2n + 1$ routing channels, connecting it to its $2n$ neighboring regular nodes as well as its local spare node. The spare node router is made of $2^n + j^n$ routing channels, connecting it to its 2^n local regular nodes and its j^n neighboring spare nodes. Each routing channel consists of one channel in and one channel out.

We next describe how the ECKN tolerates faulty nodes and faulty links. Connecting a spare node to a regular node is done to tolerate a node failure. If the spare node resides in the cluster of the faulty node, the appropriate communication channel of the spare node is merged with the communication module of the faulty node. If the assigned spare node and the faulty node belong to different clusters, a dedicated path to connect them needs to be established. Once such a path is established, due to the capabilities of the circuit-switched routing modules, the physical location of the faulty node and its assigned spare node is irrelevant. Moreover, no modification of the available computation or communication algorithm is necessary. Faulty links are bypassed by establishing parallel paths using spare links. Figure 2 illustrates reconfiguration of an ECKN with $k = 6$ and $n = j = 2$ in the presence of indicated faulty nodes and faulty links. For the sake of clarity, only active spare links are shown in the figure. Note that by utilizing the intermediate spare nodes, in effect 4 logical spare nodes are present in cluster 11. Figure 3 shows how appropriate communication channels of various spare nodes are merged with the communication module of node 22 so that the spare node $S02$ could replace the faulty node 22.

3 Reconfiguration of the ECKN

Let's define a cluster with one or more faulty nodes as a faulty cluster. Since within a cluster, the local spare node is directly connected to every regular node, the number of edge-disjoint paths between the faulty nodes of a cluster and unassigned spare nodes in other clusters is the same as the number of edge-disjoint paths between the local spare node of the faulty cluster and unassigned spare nodes. The reconfigurability of the ECKN is then a function of the number of dedicated and edge-disjoint paths, within the spare network, that can be established between the local spare node of a cluster with multiple faulty nodes and the available spare nodes in the fault-free clusters. We define the number of such edge-disjoint paths that must be constructed from a spare node as the *connection requirement* (C_R) of that spare node. For example, in Figure 2, since 3 out of 4 logical spare nodes of cluster 11 physically belong to other clusters, the C_R of the spare node $S11$ is 3. Note that the C_R of a spare node is equal to the number of faulty nodes in its cluster minus one. The following theorems establish the number of faulty nodes that can be tolerated by the ECKN.

Theorem 1 *The upper bound on the number of faulty nodes that an enhanced cluster k-ary n-cube ($k \neq 2$) can tolerate in a cluster is $2n+1$.* ■

Theorem 2 *In an enhanced cluster k-ary n-cube ($k \neq 2$), a total of $2n+1$ faulty nodes can be tolerated regardless of fault distribution.* ■

Let's group the spare nodes into three sets: S_S (set of source nodes), S_U (set of used nodes), and S_T (set of target nodes). A source node is a spare node in a cluster with multiple faulty nodes. The set S_S then represents the spare nodes with a C_R greater than 0. S_T is the set of unassigned spare nodes, and S_U consists of spare nodes that have been assigned to faulty nodes and have a C_R of 0. For example, considering only the faulty nodes in Figure 2, after assigning the local spare node to a local faulty node in each faulty cluster, $S_S = \{S11, S12, S20\}$, $S_U = \{S21\}$, and $S_T = \{S00, S01, S02, S10, S22\}$. During our reconfiguration algorithm, which is discussed later in this section, the spare nodes are dynamically assigned to the various sets. To illustrate this, suppose the C_R of a spare node $\alpha \in S_S$ is greater than 0 and there exists a dedicated path from α to $\beta \in S_T$. Consequently, β replaces a faulty node in the cluster of α. β is then called used and is assigned to S_U. Also, the C_R of α is reduced by one. If the C_R of α becomes zero, it is also marked as used and is assigned to S_U. The ECKN is reconfigured when S_S becomes an empty set.

As mentioned before, the reconfigurability of the ECKN is a function of the number of edge-disjoint paths, within the spare network, that can be established between the local spare nodes (nodes in S_S) of the clusters with multiple faulty nodes and the available spare nodes (nodes in S_T) of the fault-free clusters. However, spare nodes do not have to be interconnected as a $\frac{k}{j}$-ary n-cube. Obviously, if the spare network is a complete graph, the ECKN can tolerate $(\frac{k}{j})^n$ faulty nodes regardless of the fault distribution. Hence, the reconfigurability of the ECKN is a direct consequence of the connectivity of the topology of the spare network. Let's denote the topology of the graph connecting the spare nodes by $G=(V,E)$, where $V = S_S \bigcup S_U \bigcup S_T$ and E represents the appropriate spare links. Let the C_R of a node $n \in S_S$ be represented by $C_R(n)$ and denote the sum of the C_R's of all nodes in a set P as $\sum_{n \in P} C_R(n)$. The next theorem examines the connectivity of G as it pertains to the reconfigurability of the ECKN.

Theorem 3 *Consider a graph $G(V,E)$, where $V = S_S \bigcup S_U \bigcup S_T$. The necessary and sufficient condition for every node $n \in S_S$ to have $C_R(n)$ edge-disjoint paths to $C_R(n)$ nodes in S_T is that the minimum number of edges leaving any subset of nodes $P \subseteq V$ be greater than or equal to $\sum_{n \in (P \bigcap S_S)} C_R(n) - |P \bigcap S_T|$.* ■

Based on Theorem 3, the ECKN can tolerate a given distribution of faulty nodes provided the sum of the C_R's of any set of spare nodes (with non-zero C_R) is smaller than the number of edges leaving the set. However, for a given dimension of the ECKN, one can always find a radix that violates Theorem 3. Therefore, under the maximum number of faulty nodes, no theoretical lower bound on the number of faulty nodes per cluster can be established.

We next present our reconfiguration algorithm. An optimal reconfiguration algorithm can be developed by utilizing the maxflow/mincut algorithm. Here, optimality is measured as the ability to assign a spare node to every faulty node whenever such an assignment is feasible vis-a-vis Theorem 3. The main drawback to reconfiguration using the above algorithm is that a digraph representation of the spare network has to be constructed [5] and the spare node assignment has to be done by the host processor. To overcome these deficiencies, we next present a near optimal reconfiguration algorithm, which is called *Alloc-Spare*. The algorithm consists of three parts as specified below:

1. Early Abort: The following solvability checks are performed to determine whether the reconfiguration is feasible. If the total number of faulty nodes is greater than the number of spare nodes ($(\frac{k}{j})^n$), the reconfiguration fails. If the C_R of a spare node is greater than $2n$, the reconfiguration fails due to Theorem 1. The reconfiguration also fails if the sum of the C_R of any two neighboring spare nodes in the spare network is greater than $4n-2$ (Theorem 3).

2. Local Assignment: The local spare node of every faulty cluster is assigned to a faulty node within the cluster. If all faulty nodes are covered, the ECKN is reconfigured.

3. Non-Local Assignment: To find a set of candidate spare nodes that can be assigned to a faulty node, we utilize Lee's path-finding algorithm [6]. The algorithm begins by constructing a breadth-first search of the minimum depth d ($1 \le d \le (\frac{k}{j})^n - 1$) in the spare network from the local spare node of a faulty cluster with a non-zero C_R. If a free spare node is found, a path to the source node is formed. The algorithm guarantees that a path to a spare node will be found if one exists and the path will be the shortest possible [6]. Once a path is formed, the links associated with that path are deleted from the spare network, resulting in a new structure. If there still remain some uncovered faulty nodes, a solvability test to check the C_R of neighboring spare nodes, similar to Early Abort, is performed on the new structure and Step 3 is repeated for a higher depth d. Reconfiguration fails if $d > (\frac{k}{j})^n - 1$, which is the longest acyclic path in the spare network. ■

We implemented the algorithm Alloc-Spare for an ECKN with $k=24$, $n=4$, and $j=6$ (256 spare nodes). The simulation result for up to 256 randomly placed faulty

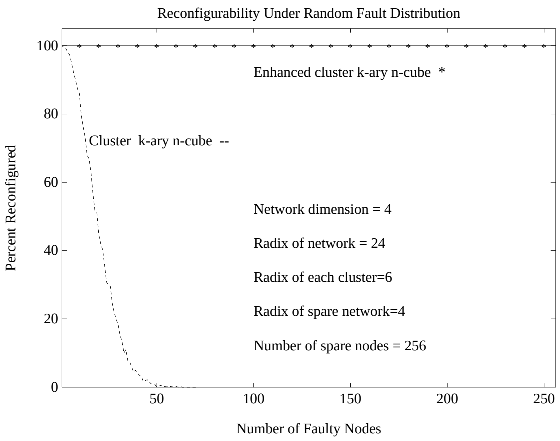

Figure 4: The ECKN under random faulty nodes

nodes is shown in Figure 4. The other plot in the figure pertains to the result of our local reconfiguration scheme, called the cluster scheme [5], whose performance is similar to the scheme proposed in [7]. The result indicates 100% reconfigurability for the ECKN under up to 256 randomly placed faulty nodes. Our simulation result further reveals that on the average about one spare link per spare node is used to reconfigure the ECKN. Hence, the spare network of the ECKN is a well connected graph even after the reconfiguration. To examine the limitation of the ECKN under random fault distribution, we next assumed that the number of faulty nodes in the ECKN is the maximum ($(\frac{k}{j})^n$). Furthermore, we assumed that each faulty cluster contains a fixed number of faulty nodes. Since by Theorem 1, a faulty cluster may have up to $2n+1$ faulty nodes, simulation runs for 1 to $2n+1$ faulty nodes per cluster were carried out. The faulty clusters were then randomly allocated in an ECKN with $k = 66$, $j = 11$, and $n = 3$ (216 spare nodes). Our simulation result indicates that nearly 100% reconfiguration is achieved (all 216 faulty nodes are tolerated) for up to 4 faulty nodes per cluster. Moreover, on the average less than half of the 6 spare links per spare node were needed to make the reconfiguration feasible. Additional simulations were carried out to measure the effect of the radix size and the dimension of the ECKN on its fault tolerance. The results suggest that for a given dimension, the radix of the spare network is inversely proportional to its reconfigurability. Furthermore, a lower dimensional spare network uses a higher percent of its spare links to reconfigure and therefore is less fault tolerant.

We implemented two algorithms to tolerate faulty links. In the first one (MPP), if any one of the spare links of a parallel path to any faulty link is unavailable, the reconfiguration fails. The second algorithm (dilation 2) considers more than one parallel path to an inter-cluster faulty link. For example, in Figure 1, if the spare link $S01 \rightarrow S02$ is not available, the faulty link $03 \rightarrow 04$ can still be replaced by the parallel path $03 \rightarrow S01 \rightarrow S11 \rightarrow S12 \rightarrow S02 \rightarrow 04$. The simulation results for an ECKN with $n = j = 4$ and $k = 24$ is shown in Figure 5. The results indicate that 90% of the time, the given ECKN can tolerate nearly 40 faulty links under MPP algorithm and nearly 50 faulty links under dilation 2 algorithm.

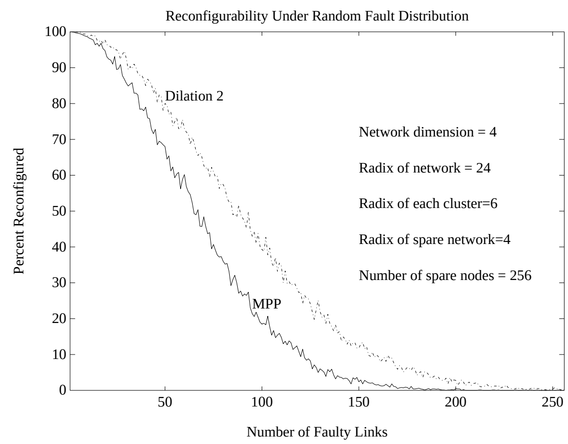

Figure 5: Tolerating link failures only

By combining the reconfiguration algorithms which tolerate node failures and link failures, a combination of faulty links and faulty nodes is tolerated [5].

References

[1] M. Alam and R. Melhem. Routing in modular fault-tolerant multiprocessor systems. In *IEEE Transactions on Parallel and Distributed Systems*, volume 6, pp. 1206–1220, November 1995.

[2] J. Bruck, R. Cypher, and C. T. Ho. Efficient fault-tolerant mesh and hypercube architectures. *Proceedings of the 22nd Annual International Symposium on Fault Tolerant Computing*, pp. 162–169, July 1992.

[3] J. Bruck, R. Cypher, and C. T. Ho. Fault-tolerant meshes with small degree. *Proceeding of 5th ACM Symposium on Parallel Algorithm and Architectures*, pp. 1–10, June 1993.

[4] S. Dutt. Fast polylog-time reconfiguration of structurally fault-tolerant multiprocessors. *IEEE Symposium on Parallel and Distributed Processing*, pp. 161–169, 1993.

[5] B. Izadi. Design of fault-tolerant distributed memory multiprocessors. *Ph.D. thesis, the Ohio State University*, 1995.

[6] C. Y. Lee. An algorithm for path connection and its applications. *IRE Transactions on Electronic Computers*, ec-10:346–365, 1961.

[7] A. D. Singh. Interstitial redundancy: An area efficient fault tolerance scheme for large area VLSI processor arrays. *IEEE Transactions on Computers*, 37(11):1398–1410, November 1988.