FAULT TOLERANCE ISSUE IN WIRED MESH, TORUS NETWORK

Santosh Kumar Singh ¹, P.K.Manjhi ²

¹Department of Computer Applications, Vinoba Bhave University, Hazaribag-825301(India)
²University Department of Mathematics, Vinoba Bhave University, Hazaribag-825301(India)

ABSTRACT:

The Parallel computers, such as multiprocessors system-on-chip (Mp-SoCs), multicomputers and cluster computers are consisting of hundreds or thousands multiple processing units and components (such as routers, channels and connectors) connected via some interconnection network that collectively may undergo high failure rates. Normally, the faulty components are coalesced into fault regions, which are classified into two major categories: convex and concave regions [1]. In this paper, we propose the occurrences of common fault patterns in torus and mesh interconnection networks which includes both Convex (L-shaped, T-shaped) and concave (L-shaped, T-shaped, +shaped, H-shaped) regions. Fault rings can be used to guide messages bypass faulty nodes/links in a fault tolerant interconnection network. However, nodes on the fault ring become hot spots, thus causing uneven distribution of the traffic loads. To avoid such traffic congestion, a concept of the balanced ring is proposed in this paper. However the algorithm presented in [2] for the formation of balanced ring has the demerits that it involves all the nodes of the network each time a balanced ring needs to form, thereby increasing overhead to the whole system. In this paper, we present an improved algorithm for the formation of balanced ring by considering only the nodes on the fault ring in contrast to all [3].

Keywords: Fault tolerance, Fault patterns, Torus, Mesh, Interconnection networks, fault ring, balanced ring.

[1] INTRODUCTION

Multicomputers, cluster- based systems and recently proposed multiprocessors system-on-chip (Mp-SoCs) as different paradigms of massively parallel computers are built with hundreds or thousands of components (processors, memories, routers, channels, and connectors) [4]. In such systems, as the number of elements increase the probability of occurrences of faults grows dramatically that may consequently affect their operation. Furthermore, due to the distributed nature of these machines, the reliability of the underlying interconnection network plays a critical design in order to keep these systems running under faulty conditions. In a network there exist two classes of faults: either the entire processing element (PE) along with its associated router can fail or just a physical link may fail. The former is referred to as a node failure, and the latter as a link failure [5]. On a node failure occasion, all physical links incident on the failed node are also marked faulty at adjacent routers [5]. Adjacent faulty nodes are coalesced into fault regions, which may lead to different patterns of failed components. Faulty regions extended by faulty components may form convex (also known as block faults) or concave shape [5, 6, 7]. In this paper, we investigate common fault patterns that may occur in a
network, particularly in mesh and torus topologies. Fault tolerance can be achieved in two ways. Firstly, it can be achieved by adding extra components to a network [5]. This kind of fault tolerance is achieved by hardware redundancy. However, adding nodes and/or links requires modifications of network topologies that may be expensive and/or difficult to implement. As a result, this method is not very popular. Secondly, fault tolerance can be achieved by designing and implementing fault tolerant routing algorithms that can find an alternative path when faulty components are encountered. This method takes advantage of inherent path diversity provided by mesh or torus topology without adding spare components, and it is more popular than the first method. There are a lot of existing fault tolerant routing algorithms proposed for mesh (e.g., [8–14]) and torus (e.g., [15–17]) networks.

Figure: 1. Illustration of the fault model.

[Figure-1] gives some illustrations examples of fault ring concept. To alleviate the congestions on the fault rings, we propose a concept of the balanced ring, which is the concentric ring for a given fault ring, after that we present Improved algorithm for the formation of the balanced ring. Our algorithm depends on the local information, i.e., only the nodes comprising the faulty ring are required to send a packet to its four neighbors to obtain their neighbors on the balanced ring and therefore it can avoid the overhead and thus, enhances the network performance. Moreover, as it depends on the local information, multiple balanced rings can be formed simultaneously for multiple fault regions.

The remainder of the survey is organized as follows. Section 2 describes the concept of fault models used in the paper, gives related background, and presents the proposed balanced ring concept. Methods of applying the balanced ring concept to the existing fault tolerant routing algorithms are also proposed in this section. Finally, Section 3 summarizes the work reported in this paper and presents possible directions for future work.
[2] SURVEY OF RESEARCH

Figure: 2. Examples of convex and concave fault regions in a 2-D torus network. (a) The l-shaped region. (b) The H-shaped region. (c) The L-shaped region. (d) The +shaped region. (e) The □shaped region. (f) The T-shaped region.
To analyze the performance of fault-tolerant systems under different conditions, it is important to identify and quantify the fault regions, which may occur in the network. These regions, which have been introduced in [5, 6, 7, 18, 19] may have different patterns and are classified as convex or concave regions. Examples of convex regions are l-shape, □-shape and concave regions are L-shape, U-shape, T-shape, H-shape, +shape which are illustrated in [Figure 2].

The mesh and torus have been popular interconnection network topologies in contemporary systems [4] due to their desirable properties, such as ease of implementation and ability to exploit communication locality to reduce message latency [5]. In addition, torus is a regular (i.e., all nodes have the same degree) and edge- symmetric network, which improves load balancing across the channels [4, 5].

**Definition 1** [16]. An R x C 2-D mesh network, denoted by M_{RXC}, consists of a set of nodes V(M_{RXC}) = {(x,y):1 <= X <= R, 1 <= y <= C} where each (x1,y1) is connected to its neighbors (x ± 1,y1) and (x1,y ± 1) if they exist. There are a total of N = R X C nodes and E = 2 X R X C - R - C channels in an M_{RXC} network.

**Definition 2** [16]. An R x C 2-D torus network, denoted by T_{RXC} and a mesh M_{RXC} topology has the same shape except the torus has wraparound channel. i.e., each node (x1, y2) is connected to its four neighbors (x ± 1 mod R,y1) and (x1,y ± 1 mod C). Therefore, the total number of channels in torus T_{RXC} is E = 2 X R X C.

In the following, we will give a brief survey on the fault tolerant routing algorithms that are most relevant to our methodology, i.e., the use of the balanced ring concept in a fault tolerant network. In [8], Chien and Kim presented a partially adaptive algorithm for mesh networks, which requires only three virtual channels per physical channel. However, their methods have to disable some nodes when faults are located on the boundaries of the meshes. Boppana and Chalasani solved this problem in [9], which uses four virtual channels. Kim and Han [10] proposed a fault-tolerant wormhole routing algorithm with four virtual channels in meshes with overlapping fault regions. Wu [11] proposed a fault-tolerant extended X–Y routing protocol, which is based on the odd even turn model and does not use any virtual channels in 2D meshes. They use extended faulty blocks (disjointed rectangles), which consist of connected disabled and faulty nodes. These algorithms [8-11] can only tolerate rectangle shaped faults. In [12], Chalasani and Boppana proposed algorithms that can tolerate some special non convex faults, such as L, T or +, with four virtual channels in meshes. In [13, 14], Zhou and Lau proposed a fault-tolerant routing algorithm with three virtual channels that can tolerate convex faults. In [15], Chalasani and Boppana developed a method to design fault-tolerant routing algorithms for torus, which can tolerate rectangle shaped fault regions by using six virtual channels. In [16], Shih improved this algorithm by reducing the number of virtual channels to three. In [17], Shih also proposed an algorithm that can tolerate convex fault regions in a torus network using four virtual channels.

The fault model is the base for the fault tolerant routing algorithms. It includes fault types, fault information and fault region. There are two types of faults: node and link faults. For example, in [Figure-1] the node (3,2) is a faulty node and the link[(1,1),(1,2)] is a faulty link. When a node fails, all physical links incident on it are considered faulty. In this paper, both types are taken into consideration. The fault information can be distributed locally and globally. However, distribution of the information globally will waste some amount of bandwidth. Some will add new signals, or even a sideband network, which will increase the network complexity.
and cost. Hence, like the algorithms discussed in [8–17], we will only consider local information based approach. Each normal node should know the status (faulty, non-faulty) of its neighbors and the links incident on it. A fault region is formed by connecting the faulty components together. For example, in [Figure-1], fault region 1(F1) includes a faulty link [(1,1),(1,2)] and F2 includes faulty nodes (3,2)(3,3)(3,4) and (4,2). Some normal nodes are disabled [(4,3)(4,4)] to form a rectangle-shaped region, which facilitates designing fault tolerant routing algorithms. A fault region F is convex if for every pair of faulty nodes n1 and n2 in F that are in the same row/column, all nodes between n1 and n2 in the same row/column are also faulty. Otherwise, it is a concave fault region. We focus on the convex fault region in this paper, although the balanced ring concept is also suitable for concave fault region. For each fault region in a mesh/torus, it is feasible to connect the fault-free components around each fault region to form a ring. This is the so called fault ring (f-ring) for that fault region. The f-ring consists of the fault free nodes and links that are adjacent to one or more components of the fault region. The f-rings of some fault regions are shown in thick lines in [Figure-1]. The fault rings are said to be overlapped if they share at least one link. For example, f-ring for F1 and F2 are overlapped at link [(2, 1), (2, 2)].

**Concept of balanced ring:**

**Definition 1** (Balanced ring). For a given fault ring, it is feasible to connect the fault-free components around the fault ring to form a bigger concentric ring of the same shape. This is called the balanced ring (b-ring) of the given fault ring, which consists of fault-free nodes and channels that are adjacent (row-wise, column-wise, or diagonally) to one or more components of the fault ring. For a given fault ring, there can exist many b-rings if the network is large enough. In this paper, we only consider one b-ring for each f-ring. Balanced rings are said to be overlapped if they share one or more channels [2].

**Definition 2** (Balanced chain). If a given faulty region includes faulty components in the network boundary, a fault chain is formed instead of a fault ring. Thus, a balanced chain is formed accordingly. Balanced chain only exists in the mesh topology. For torus networks, there are no balanced chains because of complete symmetry [2].

**Definition 3** (Partially balanced ring). If a balanced ring has to share links with its fault ring, it is called a partially balanced ring. A partially balanced ring appears when the fault ring includes components in the network boundary or two fault rings are overlapped [2].

[Figure-3] shows some examples of the definitions above. The fault ring and the balanced ring are plotted in thick and dash lines respectively. It is obvious that b-rings have the same shape with f-rings. B-ring for F1 is overlapped with b-ring for F3, because they share the same links [(6,5),(6,6)] and [(6,6),(6,7)]. The b-ring for F2 is a balanced chain because F2 is on the network boundary. The b-ring for F3 is a partially balanced ring, because it shares the link [(9,6),(9,7)] with its f-ring.
Formation of the balanced ring

We assume a single f-ring is already formed. To form a b-ring for it, each node in the networks sends a test packet to all its four neighbors to see if any neighbor is on the f-ring. Then, the function FindNeighboursInBrin() [2] is invoked to find the neighbors of the current node in the b-ring. For example, as [Figure-3] shows, for the fault region F1 there already exists a fault ring for it. Node (2, 6) obtains test packets from its four neighboring nodes, so its two neighboring nodes in balance ring are node (1, 6) and (2, 7). When each node knows its two neighbors, the b-ring is formed.

Improved algorithm for the formation of the balanced ring, i.e.

Proposed Sequential Algorithm : we assume that a single f-ring is already formed. the algorithm follows a geometric approach and is based on local computation i.e., only the links and nodes in the vicinity of faulty nodes. The basic idea of the algorithm is as follows. It starts with the uppermost right corner node on the f-ring and returns the nearest node to be kept on the b-ring. It then visits a single node on the f-ring at a time in the anticlockwise direction. Each time a node is visited, it returns either (i) the nearest node or (ii) the nearest node as well as a shape P [3] depending on whether the visited node is a non-corner node or a corner node with a LEFT turn. They are then added to the partially formed balanced ring. This is continued until the last returned node is added to the first node of the b-ring.

Distributed Algorithm: The basic idea of this algorithm is as follows, In the first phase, each node on the f-ring sends a packet to all of its four neighbors to enumerate the number of faulty
neighbors (inside the fault region) by counting non-receipts of acknowledgement. This is done in parallel for each node on the f-ring. Note that any node on f-ring can have at most two faulty neighbors. In the next phase, each node on the f-ring invokes the function Get_Node_On_B-ring() [3] in parallel to collect node(s) on the balanced ring. Each of these nodes can simultaneously collect its two neighbors and form the balanced ring in the last phase.

3. Conclusions

As technology scales, efficiency and reliability of large-scale parallel computers becomes a significant concern in the performance of these systems. One of the key issues in the design of such systems is the development of an efficient communication network that provides high throughput and low latency under different working conditions and more importantly ability to survive beyond the failure of individual components (i.e., nodes or links)[1].

To provide fault tolerance in mesh and torus networks, many routing algorithms use fault ring to guide packets bypass the fault regions. However, the nodes on the fault ring become hotspots, which will lead to early saturation of the network. To avoid this, we propose the concept of balanced ring in this paper. In this paper, we have presented a sequential and a distributed algorithm to form a balanced ring for fault tolerant routing in a two dimensional mesh. Therefore the proposed algorithms supersede the algorithm presented in [2].

A more challenging extension of our work would be to propose expressions firstly for other common fault regions in 2-D mesh and then to higher dimensional networks and other well-known topologies.

By applying the balanced ring, the f-ring-based algorithms can decide whether to bypass the fault regions along the balanced ring or the fault ring, according to the state on the f-ring. Thus, more balanced link utilization can be achieved. The results show that algorithms with balanced ring concept can yield better latency and throughput performance than those without. What’s more, it is very easy to equip the existing f-ring-based algorithm with the balanced ring concept after a minor modification of the routing function.

REFERENCES

A SURVEY ON FAULT TOLERANCE ISSUE IN WIRED MESH, TORUS NETWORK