سومین کنفرانس فناوری اطلاعات و دانش

۶ تا ۸ آذرماه ۱۳۸۶

A New Approach to Fault-Tolerant Routing in Mesh Interconnection Networks, Based on Deterministic Routing

A. H. Borhani, M. Ghaziasgar
Department of Computer Science
Islamic Azad University of Majlesi
Isfahan, Iran
Amir_h_borhani, gghmrm@yahoo.com

A. Movaghar
Department of Computer Science
Sharif University of Technology
Tehran, Iran
movaghar@sharif.edu

R. G. Cole
Department of Computer Science
Johns Hopkins University
Maryland, USA
robert.cole@jhuapl.edu

***Abstract*—Many of the calculating systems are made based on a parallel nodes architecture. In these systems, interconnection networks play a main role. Because the number of nodes is thousands, the probability of failure is high. So, developing fault-tolerant mechanisms is essential. In this paper a new fault-tolerant routing algorithm, which is based on deterministic methods, is introduced for a two dimensional (2D) mesh network. Packets are sent to their destination through XY routing algorithm and if this transmission is not possible, YX routing algorithm is applied. The XY routing algorithm nullifies offset in X direction before routing in Y direction, but The YX routing algorithm first nullifies offset in Y direction and then start routing in X direction. To evaluate this algorithm, we compare it with the Gomez method [1] which uses intermediate nodes for tolerating faults. Our results show that in environments where the fault probability is low and message generation rate is high, our method is preferred.**

Keywords-*mesh interconnection network, fault-tolerant routing algorithm, evaluation, backtracking technique, deterministic routing algorithm.*

I. INTRODUCTION

In many applications, such as protein folding, global climate modeling and galaxy interaction simulation, high processing power is needed. The required level of processing power can only be achieved with massively parallel computers, such as Earth Simulator [2], ASCI Red [3], and the BlueGene/L [4]. Having many processors, with peripheral devices such as memory, switches, and interfaces can cause a high probability of faults. Based on [1], scaling to petaflop power using present machine architectures implies a very large number of processors, of order 100,000.

Thus, due to the large number of processors, it is critical to develop robust, fault-tolerant mechanisms. In addition, failures in the interconnection network may isolate a large fraction of the machine, containing many healthy processors that otherwise could have been used. Although network components, like switches and links, are robust, they are working close to their technological limits and, therefore, they are prone to failures. Increasing clock frequencies leads to higher power dissipation; which again could lead to premature failures. Therefore, fault-tolerant mechanisms for interconnection networks are a critical design issue for large massively parallel computers [1], [12], [18], [19], [20].

Faults can be classified as transient or permanent. Transient faults are usually handled by communication protocol, using error codes to detect faults and retransmitting packets. In order to deal with permanent faults in a system, two fault models can be used: static or dynamic. In a static fault model, it is assumed that all the faults are known in advance when the machine is (re)booted. In order to implement it, once a fault is detected, all the processes in the system are halted, the network is emptied, and the management application is run in order to deal with the faulty component. The management application detects where the fault is, computes the information required by the nodes in order to tolerate the fault, and distributes the information. This fault model needs to be combined with checkpoint techniques in order to be effective. Applying checkpoints minimize the fault's impact on applications by restarting them from the latest checkpoint. In a dynamic fault model, once a new fault is found, actions are taken in