EVALUATION OF HFT TECHNIQUE PERFORMANCE USING TASK-SCHEDULING ALGORITHMS

O. A. Abulnaja

Department of Computer Science, King Abdulaziz University
Jeddah, Saudi Arabia
email: *abulnaja@kaau.edu.sa*

Abstract: *In earlier work we have introduced an efficient hardware fault-tolerant approach for reliable execution of tasks. The proposed approach called the Hardware Fault-Tolerant (HFT) approach. Also, we have introduced the concept of dynamic group maximum matching, which is used to group nodes of a graph into disjoint groups with different sizes dynamically. Furthermore, we have proposed the Dynamic Group Maximum Matching (DGMM) algorithm for finding the dynamic group maximum matching. In addition, we have proposed several hardware fault-tolerant scheduling algorithms, based on the HFT technique and the DGMM algorithm.*

In this work, we studied the effect of the HFT technique on system performance for four of the proposed scheduling algorithms. The studied algorithms are: Hardware Fault-Tolerant (FCFS + First Disagreement Graph First + First Started First), Hardware Fault-Tolerant (FCFS + First Fit First + First Disagreement Graph First + First Started First), Hardware Fault-Tolerant (FCFS + First Disagreement Graph First + Largest Group First), and Hardware Fault-Tolerant (FCFS + First Fit First + First Disagreement Graph First + Largest Group First) scheduling algorithms. Two performance metrics were evaluated. The first metric is system mean response time. The second metric is percentage of tasks of a certain type completed.

Keywords: *Performance Evaluation; Fault Tolerance; Fault Diagnosis; Task Scheduling; Networks.*

1. Introduction

In recent years, computer systems have been used in many applications ranging from critical applications to industrial process control and business applications. However, reliable execution of tasks and automatic detection and identification of faulty components where faulty outputs can jeopardize human lives or can cause loss of money are important design issues.

The first system-level fault diagnosis model in multicomputer systems was introduced in [1]. Another diagnostic system-level model was proposed in [2]. Both models assume that processors test each other by sending stimuli and waiting for a response. Due to several limitations related to this approach, other approaches were developed [3]. Among those approaches is the comparison-based model [4, 5], this approach assumes that each task is assigned to two different processors in the system for execution and a central observer compares their outputs. Then, the comparison-based approach is extended to allow the comparison of the outputs to take place in the processors themselves [6], but still the outcomes of the comparisons need to be sent to a central observer for diagnosis. In [7], the researchers allowed one of the processors being compared to be the comparator processor.