Chapter 36
A Fault-Tolerant Scheduling Algorithm Based on Checkpointing and Redundancy for Distributed Real-Time Systems

Barkahoum Kada
Department of Computer Science, University of Batna2, Batna, Algeria

Hamoudi Kalla
Department of Computer Science, University of Batna2, Batna, Algeria

ABSTRACT

Real-time systems are becoming ever more widely used in life-critical applications, and the need for fault-tolerant scheduling can only grow in the years ahead. This article presents a novel fault tolerance approach for tolerating transient faults in hard real-time systems. The proposed approach combines both checkpointing with rollback and active replication to tolerate several transient faults. Based on this approach, a new static fault-tolerant scheduling algorithm SFTS is presented. It is based on a list of scheduling heuristics which satisfy the application time constraints even in the presence of faults by exploring the spare capacity of available processors in the architecture. Simulation results show the performance and effectiveness of the proposed approach compared to other fault-tolerant approaches. The results reveal that in the presence of multiple transient faults, the average timing overhead of this approach is lower than checkpointing technique. Moreover, the proposed algorithm SFTS achieves better feasibility rate in the presence of multiple transient faults.

DOI: 10.4018/978-1-7998-5339-8.ch036

INTRODUCTION

Real-time systems have been defined as "those systems in which the correctness of the system depends not only on the logical result of the computation, but on the time at which the results are produced". They have been extensively used in many human applications like sensor networks, satellites, unmanned vehicles, and personal mobile equipment.

Fault tolerance techniques have been proposed for these systems to satisfy their constraints even in the presence of faults. Transient faults are the most common, and their number is continuously increasing due to the high complexity, smaller transistor sizes, higher operational frequency, and lower voltage levels (Djosic & Jevtic, 2013; Han et al., 2013; Paul et al., 2009; Wei et al., 2012). These faults happen for a short time and then disappear without causing permanent damage. Transient faults have become the main concern in the design of modern embedded real-time systems.

In this paper, we present a novel fault tolerance approach based on scheduling heuristic to tolerate a fixed number of transient faults. Our approach combines active replication, which provides space-redundancy, and checkpointing with rollback recovery, which provides time-based redundancy. In addition, we propose a new fault-tolerant scheduling heuristic which generates, from a given hard real-time application and a given multiprocessor distributed architecture, a fault tolerant distributed static schedule which tolerates k transient faults.

The rest of this paper is organized as follows. A brief overview of related work is provided in section 2. Section 3 describes our application model, hardware model and fault model. Section 4 presents details of our fault tolerance approach with examples. In section 5, we present our proposed static fault-tolerant scheduling heuristic. Simulation results are discussed in section 6, and finally, section 7 concludes the paper.

LITERATURE REVIEW

Extensive research has been presented to investigate the software-based fault tolerance techniques against transient faults. In the software replication technique (Girault et al., 2004; Assayad et al., 2012; Samal et al., 2013; Meroufel & Belalem, 2014) multiple replicas (active or passive) of each task are executed on different processors.

Assayad et al. (2012) proposed a new tri-criteria scheduling heuristic to minimize the schedule length, the global system failure rate and the power consumption of the generated schedule. Active replication of tasks and data dependencies is used to increase the system reliability. The primary-backup approach (passive replication) is used as a fault-tolerant scheduling technique in (Samal et al., 2013) to guarantee real time tasks constraints in the presence of permanent or transient faults. The authors proposed fault-tolerant scheduling for independent tasks using a hybrid genetic algorithm.

The replication technique is effective to tolerate spatial multiple faults (permanent or transient) and it is more preferable for safety-critical systems (Ejlali et al., 2012). However, scheduling multiple replicas of each task on different processors may not be affordable due to cost constraints (Ropars et al., 2015).

Checkpointing with rollback recovery (Han et al., 2015; Izosimov et al., 2012; Wei et al., 2012; Zhang & Chakrabarty, 2006; Kumar et al. 2015) and re-execution (Izosimov et al., 2008; Gui & Luo, 2013) are classified by Motaghi and Zarandi (2014) as time-based redundancy methods. These methods try to deal with transient faults by serial executions in the same processor of faulty task. Izosimov et al.