Co-Design Techniques for Fault-Tolerant Real-Time Systems using Imperfect Fault Detectors

1. Background

Fault-tolerant system design using active redundancy is a very challenging task that involves solving two major problems, namely finding the optimal utilization of temporal and/or spatial redundancy and the scheduling of tasks and replicas under timing constraints.

Most of the current literature assumes that all faults are detected within a certain time interval. With this assumption, each task will produce either a correct output or no output at all. The prerequisite is the existence of a perfect fault detector that achieves 100% fault coverage. Perfect fault detectors usually come with a high timing overhead. When active redundancy is concerned, there is a tradeoff about whether the available resources should be spent on implementing better fault detection or realizing more redundancy.

Beside software solutions, fault detection could also be implemented in hardware to reduce the overhead in time, e.g. using on-chip reconfigurable FPGA fabric. This not only contributes to reducing the schedule length but also allows more options for redundancy. Unfortunately, hardware fault detection increases the overall system cost. In particular, the on-chip resources are often not sufficient to implement hardware fault detectors for all tasks. Hence, it is a major design decision to select which fault detector to implement for each task and where to implement them.

2. System Model

A system consists of two parts: a heterogeneous multiprocessor platform and an application, modeled as an acyclic task graph. Besides each processor, there exists an FPGA co-processor that could be used for hardware acceleration of the fault detection components. For each task in the application, we assume that there exists a library of implementable fault detectors (both software and FPGA implementations) characterized by their detection coverage, time overhead and area overhead (for FPGA implementations only).

3. Problem Statement

Given the above system model, we formulate two optimization problems to be solved.

I. Problem 1

Constraints:

- The global deadline *D* of the application (the execution should finish before *D*).
- The total amount of on-chip reconfigurable fabric is limited by a fixed amount *A*.

Optimization objectives: Maximize reliability and minimize schedule length.

II. Problem 2

Constraints:

- The global deadline *D* of the application (same as above).
- The reliability of the whole system, given as a percentage *R*, specifying the probability that the application completes successfully.

Optimization objectives: Minimize schedule length and the total hardware area used for implementing the fault detectors.

The goals are to solve the above problems by extending an existing framework and implementing optimization algorithms for them. The reliability analysis will be given!

5. Requirements

- Strong programming and algorithmic skills.
- Knowledge of Java.

Contact: Ke Jiang, ESLAB, IDA
ke.jiang@liu.se, B 329:204

Adrian Lifa, ESLAB, IDA
adrian.alin.lifa@liu.se, B 329:206