Reliability Engineering and System Safety 180 (2018) 13–24

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

A decomposition-based reliability and makespan optimization technique for hardware task graphs☆

Reza Ramezani[a,*], Yasser Sedaghat[b], Mahmoud Naghibzadeh[b], Juan Antonio Clemente[c]

[a] Department of Software Engineering, Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran
[b] Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[c] Computer Architecture Department, Universidad Complutense de Madrid, Madrid 28040, Spain

ARTICLE INFO

Keywords:
Reliability
Makespan
Fault tolerance
Task graph
FPGA

ABSTRACT

This paper presents an approach to optimize the reliability and makespan of hardware task graphs, running on FPGA-based reconfigurable computers, in space-mission computing applications with dynamic soft error rates (SERs). Thus, with rises and falls of the SER, the presented approach dynamically generates a set of solutions that apply redundancy-based fault tolerance (FT) techniques to the running tasks. The set of solutions is generated by decomposing the task graph into multiple subgraphs, applying a multi-objective optimization algorithm to the subgraphs separately, and finally combining and filtering out the obtained solutions of the subgraphs. In this regard, a heuristic has been proposed to decompose task graphs in such a way that a high coverage of the true Pareto set is attained. The experiments show that the presented approach covers 97.37% of the true Pareto set and improves the average computation time of generating the Pareto set from 6.29 h to 81.86 ms. In addition, it outperforms the NSGA-II algorithm in terms of the Pareto set coverage and computation time. Additional experiments demonstrate the advantages of the presented approach over the state-of-the-art adaptive FT techniques in dynamic environments.

1. Introduction

Performance, reliability, efficiency, and flexibility are the main concerns of space-mission computing systems that, compared to the ground-processing ones, are faced with more radiations [1]. In addition, they may experience different soft error rates (SERs) during their missions [2], which change dynamically due to various reasons, such as solar winds [3]. For instance, it is well known that, for a given orbit, a satellite may experience unexpected fluctuations in the SER depending on factors such as the solar activity, the latitude, etc. This problem also arises in deep space missions beyond the Earth's magnetosphere, where the magnetic field of the Earth does not provide any protection against cosmic rays expelled by the Sun.

Reconfigurable computers offer the bases for a good trade-off between performance and flexibility. Therefore, they are very promising candidates to address the requirements of applications running in on-board computers, which usually feature a high level of parallelism. Typical examples are multimedia applications that perform heavy on-board computations, due to the reduced bandwidth between the flying device and the Earth [4]. The aforementioned computers support dynamic re-configurability to gain some degrees of flexibility. Despite being flexible and having a good performance, susceptibility to radiations is one of the main drawbacks of static random access memory (SRAM)-based field programmable gate arrays (FPGAs) [5], which limits their usage in space missions. Consequently, fault tolerance (FT) techniques are required to reduce the impact of the radiations. However, they impose several overhead types to the system [6]. Therefore, these techniques should be selected carefully in a way that not only the system's reliability is increased, but also its performance is not degraded significantly.

A recent study made by the authors [7] has shown that, by using optimal FT techniques, it is possible to increase the reliability of hardware task graphs running on reconfigurable computers in space missions, while degrading the performance as little as possible. This technique, which has been named *true Pareto-based technique*, employs an *exact* optimization approach to obtain the entire Pareto set of

☆ This work was supported in part by the Ministry of Science, Research and Technology of Iran and by the Spanish MCINN projects TIN2013-40968-P and TIN2017-87237-P.

* Corresponding author.
E-mail addresses: r.ramezani@eng.ui.ac.ir (R. Ramezani), y_sedaghat@um.ac.ir (Y. Sedaghat), naghibzadeh@um.ac.ir (M. Naghibzadeh), ja.clemente@fdi.ucm.es (J.A. Clemente).

https://doi.org/10.1016/j.ress.2018.07.007
Received 31 January 2018; Received in revised form 30 June 2018; Accepted 5 July 2018
Available online 06 July 2018
0951-8320/

redundancy-based FT techniques that feature different reliability and total execution time (referred to as makespan in this paper) trade-offs. This set is referred to as "true Pareto set" in the remainder of the paper. This technique exhibits exponential time complexity, since the problem size exponentially increases with the number of tasks of the running applications, which are represented as task graphs. Therefore, it is not suitable to be used in systems which are faced with unpredicted SERs during their missions.

In order to overcome said complexity, this paper presents a new technique that is based upon the decomposition of the task graphs into several subgraphs and carrying out the technique presented in [7] to separately optimize the reliability and makespan of each one of them. Experimental results will demonstrate that this technique, named *decomposition-based Pareto set technique* optimizes the reliability and makespan of hardware task graphs, in a much more efficient way than [7], while achieving a high coverage of the true Pareto set of solutions. Thus, this technique is suitable for applications running in FPGA-based systems that operate in harsh environments with dynamic and unpredicted SERs. In order to perform such task-graph decomposition, this technique implements a new heuristic, called *Least Overlapping*, which is based upon detecting the subgraphs whose execution least overlaps with the subsequent subgraphs when FT techniques are applied to them.

The experimental results demonstrate the efficient computation time of the presented approach and the high true Pareto set coverage obtained by the *Least Overlapping* heuristic. The experiments also reveal that the presented approach outperforms the well-known stochastic optimization method *non-dominated sorting genetic algorithm II* (NSGA-II) [8] in terms of the true Pareto set coverage and computation time. This method extends naive genetic algorithms by using an elitism mechanism for building the solutions of the next generation by considering the rank of the generated solutions in the Pareto set. Additional experiments, in environments with dynamic SERs, show the usefulness of the presented approach over the state-of-the-art adaptive FT techniques, in terms of accuracy, computation time, and reliability and makespan improvement.

The rest of the paper is organized as follows. Section 2 introduces some related studies. Section 3 formulates the presented problem and presents an illustrative example. Section 4 presents the general methodology and foundations of the presented approach. Section 5 elaborates the task graph decomposition heuristic and extends the illustrative example. Section 6 gives the experimental evaluations and results, and finally Section 7 concludes the paper and offers suggestions for future work.

2. Related work

As it will be shown in the next section, in this paper, an active redundancy-based FT technique is used to increase the reliability of tasks. In addition, a task graph is executed correctly if all its tasks are executed without any failure. Therefore, from the point of view of their reliability, in this work it is assumed that applications run as a parallel-series system [9]. Several researches have been done on maximizing the reliability of parallel-series systems while minimizing other parameters, such as: weight, volume, makespan, power consumption, which is referred to as *redundancy allocation problem* (RAP). For example, the work presented by [10] maximizes the reliability of a parallel-series system while minimizing its cost, using NSGA-II. A similar method is used to estimate and optimize the mean time to failure (MTTF) of these systems with the assumption of using standby and active FT techniques [11]. Nevertheless, the architecture of these systems differs from FPGA-based ones in the sense that in the parallel-series system of RAP, each subsystem is considered as a disjoint processing unit and has just one functionality. Thus, no scheduling and placement is taken into account in those methods. In this work, there is an FPGA that is shared among all the tasks and runs the task graphs using a scheduling algorithm enhanced with a task placement mechanism.

Several studies have also focused on improving the reliability of FPGA-based designs without paying attention to the adverse effects of applying FT techniques. These studies can be categorized into three groups of mitigation approaches: design-based, placement- and routing-based, and recovery-based methods.

Design-based methods involve redundancy, which can be applied at different granularities and different levels of design [12]. Some studies have investigated the application of redundancy-based FT techniques in increasing the reliability of FPGA-based space-computing systems [13], Cloud platforms [14], and embedded processors [6]. The *Adaptive FT technique* presented by [15] is another example of design-based methods that uses different redundancy-based FT techniques for different ranges of SERs.

Placement- and routing-based FT techniques increase the reliability of an FPGA-based design by adapting traditional place & route techniques at different design phases. For example, an interesting technique has been presented in [16], which manages the signals between functions in such a way that multiple errors affecting two different connections are not possible. In a similar approach, [17] studies both fault occurrence and error propagation probabilities to propose a reliability-oriented placement and routing algorithm. Anyway, as indicated by [18], all these techniques can be used in combination with other design-based methods to increase the reliability of the designs.

However, the aforementioned techniques cannot prevent fault accumulation at run-time in FPGAs. Some recovery-based methods are especially designed for FPGA-based reconfigurable computers to prevent fault accumulation in configuration memory by recovering the faulty cells [19]. For example, the study in [20] determines different scrubbing rates for different circuits, based on their failure rate, in such a way that the system reliability is maximized. In the scrubbing process, the information of the reconfiguration memory is periodically refreshed. A new approach in increasing system reliability with respect to power optimization using scrubbing has been studied in [21]. Frame-level redundancy scrubbing (FLRS) is another technique, presented by [22], which applies frame-level scrubbing to triple modular redundancy (TMR) designs to correct accumulated upsets.

Combining design-based and recovery-based methods is very effective for mitigating soft errors in FPGA-based systems. For example, the work by [23] employs a redundancy-based approach implemented with spare units, and the work by [24] combines the TMR technique with recovery ones to provide self-repair capability.

In any case, most of the studies of FT techniques in FPGA-based computers assume that all the existing hardware tasks can be placed simultaneously on the device, and therefore they have not addressed the problem of task scheduling. In fact, FPGA-based systems have limited resources and scheduling techniques are required to manage the execution of the tasks [25].

However, very few studies have considered both scheduling and FT techniques simultaneously. These methods aim at increasing the system's reliability whereas a given performance is guaranteed as well. For example, the *Early-fetch* technique, presented by the authors [26], increases the reliability of a hardware task graph without degrading its performance. Similarly, a run-time system for managing hardware and software tasks has been presented in [27] to explore different trade-offs between reliability and performance of FPGA-based systems. In [28], a real-time fault-tolerant scheduling algorithm has been proposed that uses a sufficient schedulability test condition to schedule hybrid hardware/software tasks that can tolerate f_i faults during their execution. In this regard, we have investigated the performance overhead of different FT strategies for different real-time scheduling algorithms in FPGA-based computers [29]. Similarly, a reconfigurable FT framework has been presented in [30], which uses a scheduler to combine task computation time and system fault rate to determine the optimal FT mode for the tasks.