



# Communication and migration energy aware task mapping for reliable multiprocessor systems



Anup Das\*, Akash Kumar, Bharadwaj Veeravalli

Department of Electrical and Computer Engineering, National University of Singapore, Singapore

## HIGHLIGHTS

- A two-step methodology is proposed for mapping applications on multiprocessor systems.
- In first step communication energy aware design space exploration is performed.
- In second step, mapping is performed to minimize communication and migration overhead.
- A significant energy savings is reported using the proposed methodology.

## ARTICLE INFO

### Article history:

Received 4 January 2013

Received in revised form

2 May 2013

Accepted 17 June 2013

Available online 26 June 2013

### Keywords:

Fault-tolerance

Heterogeneous MPSoCs

Design space exploration

Task mapping

Mixed integer quadratic programming

## ABSTRACT

Heterogeneous multiprocessor systems-on-chip (MPSoCs) are emerging as a promising solution in deep sub-micron technology nodes to satisfy design performance and power requirements. However, shrinking transistor geometry and aggressive voltage scaling are negatively impacting the dependability of these MPSoCs by increasing the chances of failures. This paper proposes an offline (design-time) task remapping technique to minimize the communication energy and task migration overhead of an application mapped on a heterogeneous multiprocessor system for all processor fault-scenarios. The proposed technique involves two steps – (1) Communication Energy driven Design Space Exploration (CDSE) to select an initial mapping and (2) Communication energy and Migration overhead aware Task Mapping (CMTM) for different fault-scenarios. The CDSE is formulated as a Mixed Integer Quadratic Programming (MIQP) problem and solved using an energy-gradient based heuristic. The CMTM problem is solved using a fast heuristic with the starting mapping selected using CDSE step. The proposed two steps technique is compared with state-of-the-art approaches through rigorous simulations with synthetic and real application graphs. Results demonstrate that the proposed CDSE reduces design space exploration time by 99% with a maximum variation of 5% from the optimal solution obtained by solving the MIQP problem directly. Further, the proposed CMTM reduces communication energy by an average 35% and migration overhead by an average 20% for all single and double fault-scenarios as compared to the existing fault-tolerant techniques. The CMTM also achieves over 30x reductions in execution time for large problem sizes with a maximum deviation of 15% from the minimum communication energy achievable with the given application on a given architecture. For streaming multimedia applications, the proposed technique delivers 50% higher throughput per unit energy as compared to the existing approaches.

© 2013 Elsevier B.V. All rights reserved.

## 1. Introduction

To accommodate the ever increasing demands of applications and for the ease of scalability, multiprocessor systems-on-chip (MPSoCs) are becoming the obvious design choice in current and future technologies [1]. With reducing feature size and increasing transistor count, MPSoCs are becoming susceptible to faults [2,3]. There are three classes of faults studied for Integrated Circuits

(ICs)—permanent, intermittent and transient. Permanent faults are irrecoverable damages to the circuit caused by phenomena such as electro-migration, dielectric breakdowns, broken wires etc. These faults are caused during manufacturing or during the product lifetime due to component wear-outs. Behavior of a system under permanent faults is time invariant. Intermittent faults are also hardware faults occurring frequently but irregularly over a period of time due process, voltage and temperature (PVT) variations. Intermittent faults usually persist for few cycles, if not for a few seconds or more. Transient faults are single event upsets occurring due to alpha or neutron particles from cosmic radiations. This paper focuses on permanent and intermittent faults which are jointly referred as *hardware faults* throughout the rest of this paper.

\* Corresponding author. Tel.: +65 90031395.

E-mail addresses: [akdas@nus.edu.sg](mailto:akdas@nus.edu.sg) (A. Das), [akash@nus.edu.sg](mailto:akash@nus.edu.sg) (A. Kumar), [elbv@nus.edu.sg](mailto:elbv@nus.edu.sg) (B. Veeravalli).

Hardware faults are traditionally tolerated using redundancy-based designs [4]. However, this is only applicable for hardware–software co-design methodology [5] where optimization results determine MPSoC architecture. Software techniques like task remapping [6–8] have shown significant promise to tolerate hardware faults especially for platform-based MPSoC design [9] where fault-tolerance needs to be incorporated on a fixed architecture. Existing platform-based fault-tolerant task-migration research studies have focused on minimizing migration overhead [6] and load balancing [7] or maximizing the reliability of a system [8]. These techniques provide no guarantee on the application communication energy.

Another research direction for multiprocessor systems is concerning energy consumption. This is partly due to the slow growth in battery technology over the past decades. Researchers have shown that carefully selecting an application task mapping can significantly reduce task communication energy [10–13] which constitutes a large fraction ( $\approx 60\%$  according to [11]) of the overall energy consumption. This is orthogonal to dynamic voltage or frequency scaling capabilities of an MPSoC which can further reduce the task computation energy. These works do not target fault-tolerance.

When one or more processors of an MPSoC become faulty, tasks from these processor(s) are migrated to new locations (functional processors). Tasks migrated further away from their dependent tasks consume more communication energy per iteration of the application graph. Recently, there are studies to integrate fault-tolerance and energy minimization [14–16]. However, either they are limited to transient faults or they do not address different processor fault-scenarios. This paper attempts to solve the following problem. Given a heterogeneous MPSoC architecture and a set of high performance applications modeled using directed graph.

- Generate a starting mapping of an application on the multiprocessor platform such that task communication energy is minimized and
- Generate task mappings for different processor fault-scenarios with the joint objective of minimization of migration overhead and task communication energy.

In a recent work [17], these challenges are solved for multimedia applications running on homogeneous MPSoCs. Task mappings are generated exhaustively and evaluated. The minimum communication energy mappings are retained for the optimization of migration overhead for different fault-scenarios. One limitation of this technique is that the application domain is restricted to multimedia programs running on homogeneous processors. Additionally, as the number of tasks and/or processors increase, there is an exponential growth in the number of mappings which can lead to longer design cycles.

*Contributions:* following are the key contributions.

- A Mixed Integer Quadratic Programming (MIQP) formulation of the Communication Energy based Design Space Exploration (CDSE).
- A energy-gradient based heuristic to minimize the design space exploration time.
- A communication energy and migration overhead aware fast heuristic for task mapping (CMTM) for different processor fault-scenarios.
- Consideration of streaming and non-streaming applications on heterogeneous MPSoC.

The CDSE is solved at design-time to generate a starting mapping of a given application on the given platform which minimizes the task communication energy. Starting from this mapping, the CMTM generates a set of task mappings for all fault-scenarios

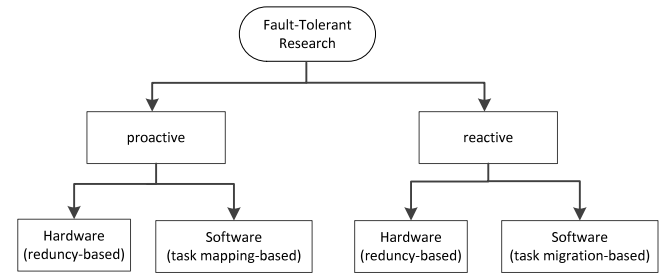


Fig. 1. Taxonomy of fault-tolerance research.

which are Pareto-optimal in terms of communication energy and migration overhead. These mappings are stored in a table for use at run-time as and when faults occur. Experiments are conducted with synthetic and real application graphs, both from streaming and non-streaming domain demonstrate that the proposed energy-gradient heuristic is able to reduce analysis time by 99% with a maximum of 5% deviation from the communication energy optimum value obtained by solving the MIQP problem directly. Moreover, the overall fault-tolerant technique minimizes communication energy by an average 35% and migration overhead by 20% as compared to the existing fault-tolerant techniques.

The rest of the paper is organized as follows. A brief overview of the prior research works is provided in Section 2 followed by a motivational example in Section 3. Next, the proposed fault-tolerant design methodology is highlighted in Section 4. Different components of this methodology are discussed in Sections 5 and 6. Experimental setup and results are discussed next in Section 7 and the paper is concluded in Section 8 with key future directions.

## 2. Related works

Reliability and energy efficiency are the two primary optimization objectives for multiprocessor systems in deep sub-micron technologies. Research works for both the objectives are carried independently over decades until recently, when efforts are directed towards joint optimization of energy and fault-tolerance. This section provides an overview of some key research results for each of these research directions.

### 2.1. Fault-tolerance related studies

There are two broad categories of hardware fault-tolerance research—*proactive* and *reactive* as shown in Fig. 1. The *proactive* techniques prevent (or delay) the occurrence of failures [8,18,19]. The *reactive* techniques deal with task migration after faults have occurred [6,7,20–23]. Task remapping decisions can be pre-computed at compile-time analyzing all possible fault-scenarios (*static*) or can be decided at run-time as and when faults occur (*dynamic*).

Static task migration techniques compute task mapping decisions at compile-time for different fault-scenarios [20–22]. The band and band reconfiguration technique of [20] minimizes migration overhead while deciding new locations (cores) for the tasks on faulty cores. Communication energy is not guaranteed in this technique. The technique of [21] maximizes application throughput but provides no guarantee on the task migration overhead and communication energy. The authors in [22] established the importance of migration overhead and throughput for multimedia applications and proposed a joint optimization technique for multimedia MPSoCs. Communication energy is not optimized in this technique either. Moreover, the number of mappings explored at compile-time grows exponentially with the number of tasks and/or cores making this technique computationally in-feasible for large scale computing.

Download English Version:

<https://daneshyari.com/en/article/6873675>

Download Persian Version:

<https://daneshyari.com/article/6873675>

[Daneshyari.com](https://daneshyari.com)