Microelectronics Journal 46 (2015) 1371–1379

Contents lists available at ScienceDirect

Microelectronics Journal

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

A role-changeable fault-tolerant management strategy towards resilient NoC-based manycore systems

Zixu Wu, Fangfa Fu, Yu Lu, Jinxiang Wang*

Microelectronics Center, Harbin Institute of Technology, Harbin 150080, China

ARTICLE INFO

Article history:
Received 4 April 2014
Received in revised form
25 July 2015
Accepted 17 September 2015
Available online 4 November 2015

Keywords:
Fault-tolerant
Management strategy
Resilience
Network-on-chip
Manycore system

ABSTRACT

In hierarchically managed network-on-chip (NoC) based manycore systems with over one thousand processor cores, resource management is critical to efficient operation of the whole system. Meanwhile, the fault-tolerant design of the management structure is as important as the fault-tolerant design in the micro-structural level and the system level, which should require more attention towards constructing resilient manycore systems. This paper presents RCFTM, a hierarchical agent-based role-changeable fault-tolerant management strategy based on the modified role-changeable management framework for resilient NoC-based manycore systems. It is a distributed and adaptive strategy which can reconstruct the management hierarchy dynamically and it is capable of providing an intrinsic ability to tolerate various damages to the management hierarchy due to permanent core faults. The fault-tolerant capability of the RCFTM strategy is first evaluated with MATLAB for a large scale manycore system. Results show that the RCFTM strategy has better fault-tolerant capability than the conventional replacement strategy which relies only on backup cores. Then, the RCFTM strategy is implemented in C on a 5×5 NoC-based manycore system, where a full-system simulation platform is utilized. Experiments show that, with 20 K bytes ROM and 35.6 K bytes RAM footprint for the RCFTM strategy on each agent, the system can successfully tolerate both single and multiple core faults. Results also show that the RCFTM strategy only introduces less than 1.48% computing overhead of a working agent while the system is fault-free.

1. Introduction

With the ever increasing integration capacity of transistors and the ever increasing demand for computation throughput, today, a large number of processor cores are integrated on a single chip. For example, the prototype Teraflops fabricated by Intel has 80 tiles on chip [1], and the commercial processor TILE-Gx8100 has 100 identical general purpose processor cores [2]. In the future, a chip with over one thousand processor cores will be possible [3,4]. Typically, these chips employ a network-on-chip (NoC) based manycore architecture, where small and simple processor cores are interconnected by the on-chip network. With so many cores on chip, resource management is critical for maximizing the system performance while reducing total power consumption. The conventional centralized management with only one processor core serving as the manager is not scalable since the manager core could become the performance bottleneck when the number of cores of the manycore system increases. To overcome this limitation an efficient approach is to adopt the hierarchical management strategy [5–7].

When designing manycore systems, fault tolerance is one of the most concerns [8–10]. To build a resilient hierarchically managed manycore system, not only the fault tolerance of links, routers and cores but also the fault tolerance of the management hierarchy should be given great attention. Because if the management hierarchy fails, no matter how well the system can tolerate faults in other aspects, such as micro-structural faults, the whole system would still function incorrectly due to the lack of system management. Therefore, ensuring the availability of system management is one of the premises to guarantee the reliable functionality of the system. Although a large number of researches have been performed on NoC fault tolerance, only a few address the problem of fault-tolerant resource management in hierarchically managed manycore systems. Most of the approaches rely on the use of multiple backups, but they face a serious problem of the exhaustion of backups. In this paper, we propose a hierarchical agent-based role-changeable fault-tolerant management strategy, RCFTM, towards constructing resilient manycore systems. This strategy attempts to keep the manycore system alive (i.e. continuing functioning) by allowing cores to change their levels in the management hierarchy dynamically according to remaining resources.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 presents our motivation.

* Corresponding author.

http://dx.doi.org/10.1016/j.mejo.2015.09.010
0026-2692/

Following that, a role-changeable fault-tolerant management strategy is presented in detail in Section 4. And evaluations of our strategy are presented in Section 5. Finally, Section 6 summarizes the paper and provides future directions.

2. Related work

Chou et al. [11] presented a system-level fault-aware resource management technique which covers task migration from fault cores to spares and fault-tolerant mapping. However, they mainly focus on the fault tolerance of computational cores and do not discuss relations among distributed managers. Schley et al. [12] studied the problem of privileged node distribution and its impact on privileged node fault tolerance in NoCs. However, since the distribution of privileged nodes is fixed at the time of design and cannot change during run-time, this strategy is not capable of handling a situation when all the privileged nodes fail. To add system level intelligence to the traditional NoC design for system self-awareness and fault tolerance, Rantala et al. [13] presented a layered agent-based reconfiguring concept. Indeed, the utilization of autonomous agents provides a good fault-tolerant potential. But the presented system control concept relies on the rules that the lower level agents are instructed by the upper level agents and inform the upper level about degraded performance or faulty functioning, which has a drawback of lower level missing instructions when all upper level agents and backup agents fail. Besides, the assumption of a cell agent with a permanent failure in itself sending a failure message to the cluster agent might be false because the message might not be sent out due to the cell failure. A similar yet more detailed discussion is presented by Guang et al. [6]. In their work, monitor services and fault-tolerant schemes at each agent level are carefully considered to enable system autonomy and the adaptivity to variations and faults. The fault-tolerant strategy for platform and cluster agents, however, still depends on backups. Although a large number of backups could be added, the fact is that the number of faults occurring in a system does not reduce because of an increased number of backups. Thus this strategy, again, cannot handle the situation when all platform and cluster agents fail.

To eliminate the dependence of physically redundant cores and to ensure that the system is still manageable even when all manager agents fail, in this paper, we consider the problem of tolerating agent faults from the perspective of the whole management hierarchy. We propose a hierarchical agent-based role-changeable fault-tolerant management strategy, RCFTM, towards constructing resilient manycore systems. When some cores fail, these mechanisms keep the manycore system alive (i.e. continuing functioning) by allowing surviving cores to change their levels in the management hierarchy and reconstruct the management hierarchy dynamically according to the remaining resources and predefined rules.

It is worth noting that the failures targeted in this paper are permanent core faults. Since many effective fault-detecting and fault-tolerant schemes have been proposed to tackle with transient and permanent faults in the network interfaces, on-chip switches and links [14], the communication infrastructures of the NoCs in our work are assumed to be reliable. Although real faults manifest themselves in many forms and hardware as well as software fault-tolerant strategies can be utilized to keep a core functioning, there is always a worst case when the core cannot maintain functioning after trying multiple strategies. From the viewpoint of management, a core is not manageable if it cannot maintain functioning. The faulty core should be stopped before it does further harm to the management of the whole system. Therefore, core faults are assumed to follow the fail-stop model [15] in our research work. We also assume that the faults can be detected by hardware or software mechanisms (such as BIST or SBST). This assumption is based on the online fault diagnosis and online error detection researches in Refs. [15,16].

3. Motivation

As pointed out earlier in [3], the manycore architecture is a resilient microarchitecture. And if the system can continually detect errors, isolate faults, confine faults and reconfigure the hardware, there is no need for one-time factory testing, since it is capable of testing and reconfiguring itself throughout its lifetime. This resilient characteristic is especially important to the application of manycore systems in space tasks [17,18]. Generally, the term “resilience” means the intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations under both expected and unexpected conditions [19]. As commented in [20], it emphasizes the ability of a system to continue functioning, rather than simply to react and recover from disturbances. Therefore, a resilient manycore system should have an intrinsic ability to continue functioning in various fault cases.

Conventionally, redundant or spare cores are introduced on chips to provide backups for working cores (i.e. the cores that are currently in operation). When some of the working cores fail during system operation, backup cores can be used to replace faulty cores during run-time. In a hierarchical software manager based homogeneous manycore system, all cores are identical. Although the designation as managers can be done at the software level when tasks are assigned to different cores, the management hierarchy fault tolerance is not merely a matter of assigning the management tasks to backup cores or other healthy cores. It is the strategy of assuring an appropriate management hierarchy rather than the procedure of reassigning management tasks that matters the most when taking resilience into consideration. Generally, the replacements of faulty cores by backup cores are initiated by manager cores, not computational cores. Thus if manager cores fail, higher-level managers would be responsible for making the replacement decision. When the highest manager core fails, however, the whole system fails, since no other managers could supervise it. Certainly, dedicated backup cores can be added to the highest manager, but the number of backups is still limited. And when all manager cores fail, even if all computational cores are healthy, the system still fails because the management hierarchy crashes. Therefore, the key resilience problem of the conventional backup core based fault-tolerant strategies lies in the only dependence on backup cores and the fact that the cores cannot change their roles in the management hierarchy. However, if the roles of cores in the hierarchy can be changed dynamically, new managers would be selected and the system would continue functioning based on the remaining computational cores.

4. Role-changeable fault-tolerant management strategy

4.1. Role-changeable management framework

In manycore systems, each processor core with software running on it can be considered as a single autonomous agent. And all these agents are organized with a hierarchical management structure. The management structure targeted in this work is illustrated Fig. 1. Similar to the hierarchical framework in previous researches [6,7], the proposed role-changeable structure consists of a level-1 agent, some level-2 agents, multiple level-3 agents and many level-4 agents. The lowest level (i.e., level-4 in this case) agents are often designated as computational agents which contribute most to the computing power of the whole chip, and agents on other levels not only perform computation but also are responsible for regional resource management. The level-1 agent serves as a global manager