Microprocessors and Microsystems 38 (2014) 552–564

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

Reliability improvement in private non-uniform cache architecture using two enhanced structures for coherence protocols and replacement policies

Mohammad Maghsoudloo, Hamid R. Zarandi *

Department of Computer Engineering and Information Technology, Amirkabir University of Technology (Tehran Polytechnic), Iran

A R T I C L E I N F O

Article history:
Available online 26 December 2013

Keywords:
Cache reliability
Temporal vulnerability factor
Cache coherence protocols
Cache replacement policies
Many-core processors

A B S T R A C T

In this paper, a comprehensive study is first conducted to investigate the effects of cache coherence protocols and cache replacement policies on the characteristics of NUCA in current many-core processors. The main focus of this study is to analyze the effects of coherence protocols and replacement policies on the vulnerability of caches. The outcomes of this analysis indicate two facts: (i) Differences in handling write operations play an important role to make distinction in favor of or against a cache coherence protocol; (ii) Near-optimal solutions for replacement problem, aimed at enhancing the performance, can also make positive influence on reduction of cache vulnerability factor. Based on the results of first step, two schemes are introduced to enhance the reliability of caches by applying some modification on the structures of cache coherence protocols and cache replacement policies. The first scheme tries to manage sharing of the dirty data items among different same-level caches. The second helps to give priority and more opportunity to old dirty blocks than clean blocks for replacement. The proposed schemes reveal about 18% improvement in MTTF, with negligible performance, bandwidth and energy consumption overhead compared to previous cache structures.

1. Introduction

While chips continue to integrate more processing cores, scalability of shared resource architectures is increasingly playing important roles in performance and energy consumption. Tiled many-core chips are emerging as the architecture of choice to provide scalable structures for shared resources, such as interconnection network, and shared L2 cache [1,2]. In light of scalability limitations for conventional interconnection network architectures (such as shared bus and crossbar switches), existing many-core chips, have featured a mesh-based interconnection network [3,4]. Furthermore, tiled many-cores give rise to varying access latencies between the cores and the cache slices spread around the die, naturally leading to a Non-Uniform Cache Architecture (NUCA) of the on-chip L2 cache [5].

Furthermore, recent studies have shown that as designers have tried to integrate more of the cache memory hierarchies onto the processing die, they are particularly most vulnerable on-chip components. In today's microprocessors, more than 60% of chip area is occupied by the different levels of caches, and they are more likely to be exposed to the soft errors [6–10]. Most soft errors result from energetic particle strikes induced by high-energy neutrons of cosmic rays, and alpha particles of decaying radioactive impurities in packaging and interconnect materials [6]. Another problem of cache memories in current technologies is the increased likelihood of single-event Multi-Bit Upsets (MBUs) in their arrays [6,7]. The MBUs are the consequence of a single particle strike, which can flip multiple bits, simultaneously [6,7].

While lightweight detection mechanisms can cheaply solve the problem of detecting MBUs, the ability of MBU correction cannot be easily provided [7]. The dirty data in caches are more susceptible against MBUs, because they have no valid copies in the other levels of the cache memory hierarchy. The inefficiency of previous correction (recovery) mechanisms in facing MBUs for dirty data items have been shown previously [7]. Enhancing the capability of previous correction techniques (for covering MBUs) poses significant performance, energy and area overheads [6].

In order to derive a cost-effective mechanism for enhancing cache reliability, this paper proposes two schemes to handle the problem of shared and private dirty data vulnerability. The first scheme focuses on the problem of shared dirty data vulnerability, and the second one concentrates on the problem of private dirty data vulnerability.

* Corresponding author. Tel.: +98 21 6454 2702.
E-mail addresses: m.maghsoudloo@aut.ac.ir (M. Maghsoudloo), h_zarandi@aut.ac.ir (H.R. Zarandi).

0141-9331/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.micpro.2013.12.003

The basis of idea of the first scheme is to use the natural features of private distributed L2, such as data replication and migration, in the context of cache coherence protocols. First, this paper shows that different load and utilization of cores in many-core chips leads to different cache memory usages among tiles. Developing parallel software, takes the advantages of Thread-Level Parallelism (TLP), can be challenging in today's programming languages. Given the difficulty of extracting parallelism, cores on the many-core processors are not being uniformly leveraged [11]. Our simulation results reveal that L2 cache slices in heavily used cores is forced to evict a block, for 99.6% of the cases, where free spaces exist in the L2 cache slices of the other tiles. In fact, there are some free spaces in the entire L2 cache which are not used at almost any time during execution by the other cores. This problem occurs due to the structure of private L2 cache, in which, each tile's L2 cannot service the requests from the other tiles through interconnect. The first proposed scheme tries to exploit these free spaces to store valid copies for dirty data items. These key points are extracted from an analysis that shows the cache coherence protocols can play impressive roles to enhance or to undermine the level of reliability for data caches. This feature determines the lifetime duration of cache blocks [12] and the events, which can affect the vulnerability of cache. Various models and protocols have been devised for sustaining the cache coherence for write-back caches, such as write-invalidate (MESI, MOESI and MESIF), and write-update (Dragon and Firefly) protocols [13,14]. Due to the negative effects of write-update protocols on the interconnection network and memory bandwidth, write-invalidate protocols are more popular than write-update ones [13,14]. This paper shows that, as these protocols imply different characteristics in terms of performance and energy consumption, they also differ with respect to cache Temporal Vulnerability Factor (TVF) and Mean Time To Failure (MTTF). Since almost all of well-known coherence protocols have a similar behavior in handling the private and shared clean data items, differences in handling private and shared dirty memory blocks make distinction among coherence protocols in favor of the MOESI protocol. Adding the Owned state in the structure of the MOESI, which provides the ability of sharing dirty data items among same-level caches, is associated with improved TVF.

The second proposed scheme reduces the negative impacts of private dirty blocks on the reliability via applying some modification on the structure of replacement policies. As the use of set associative mapped caches increases in current architectures, the importance of cache replacement policies is also increased [15–17]. The Commercial-Off-The-Shelf (COTS) microprocessors employ various policies for replacement mechanism such as Random [12], LRU (Least Recently Used) [18], First-In-First-Out (FIFO) [19], and two versions of PLRU (Pseudo LRU) [20]. This paper indicates that near-optimal replacement policies, aimed at enhancing the hit ratio (performance), can also make positive influence on reduction of cache vulnerability factor. Therefore, the LRU-based policies imply better characteristics in terms of TVF and MTTF, as well as hit ratio, compared to the other policies. While the LRU policy has high implementation cost and requires complex logic, two heuristics, presented as alternatives for LRU, lead to better overall results. These policies reduce amount of time, cache blocks held in the phase between the last write and replacement; and subsequently lead to lower TVF compared to the other policies. The second proposed scheme tries to give more opportunity to dirty blocks for replacement compared to the clean blocks. When a dirty block is selected as a candidate for replacement, the policy cannot change the candidate except under the circumstance that the new candidate block is also dirty.

To evaluate the proposed design decisions, SPLASH-2 benchmarks suite [21] are utilized. These benchmarks were run on a functional simulation infra-structure, SIMICS [22], used with an extended version of Multi-facet GEMS [23]. The results of experiments reveal that the proposed schemes imply 18.3% improvement in MTTF of data caches, with 2.6% performance, 5.3% bandwidth and 2.5% L1 cache energy consumption overhead compared to previous works and cache structures. The other important point is that the effects of two proposed schemes are approximately cumulative on MTTF. While these schemes concentrate to reduce two different vulnerable phases during lifetime of a cache block (the phase between a write and its next reads, and the phase between last write and replacement), their effects on TVF are almost independent of each other.

The rest of paper is organized as follow: Section 2 describes the background and terminologies. The proposed schemes are explained in Section 3. Section 4 shows the experimental results. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. The MIT RAW architecture

The Tile architecture has its origins in the RAW research processor developed at MIT and later commercialized by Tilera, a start-up founded by the original research group [24]. The key differentiating structure of the tiled architecture is the on-chip interconnection network and the L2 cache architecture. As the name implies and depicted by Fig. 1, the chip is structured as a two-dimensional array of tiles, where each tile contains a processor core with dedicated L1 caches, a slice of the L2 cache, a slice of the directory and two network interfaces connecting the node to the Network on Chip (NoC). The L2 slice can be either a part of a shared L2 cache or a private L2 for the local core. In case of a shared L2, cache blocks are address-interleaved among the L2 slices. In case of private L2s, each core accesses its own L2 tile [1,24]. Due to higher hit rate, private L2 cache organizations have been more concentrated by industry and academia than shared L2 [25–27]. Our contribution in this paper is also a novel NUCA design for tiled many-core based on private partitioning, because of following advantage:

- Private L2 caches have the advantage that most L1 misses can be handled locally, so the number of remote on-chip L2 cache accesses is reduced [26,49].
- Private cache based designs lend themselves to easier implementation of performance isolation, priority and QoS, which traditionally have been assumed or needed by applications and operating systems [26,50].
- Instead of building a highly associative shared cache to avoid inter-thread contention, the same set-associativity is available for an aggregate cache formed by the private caches, each with much lower set associativity, thus reducing power, complexity and latency overhead [26].
- The bandwidth requirement of the cross chip interconnection network is generally much lower with private L2 caches, and this leads to a simpler and potentially higher performing network [26].

2.2. Related work

The solution space of existing techniques can be classified into process-, circuit- and system-level solutions.

2.2.1. Process-level techniques

The key solution of process-level techniques that can reduce the soft error rate is modifying the structure of SRAM cell at the layout-level. Deep N-well technology and Silicon-On-Insulator (SOI) are two common methods to provide isolation from harmful particle