CCIT Report #607 December 2006

The Power of Priority: NoC based Distributed Cache Coherency

Evgeny Bolotin, Zvika Guz, Israel Cidon, Ran Ginosar and Avinoam Kolodny
Electrical Engineering Department,Technion - Israel Institute of Technology, Haifa 32000, Israel

ABSTRACT

The paper introduces Network-on-Chip (NoC) design methodology and low cost mechanisms for supporting efficient cache access and cache coherency in future high-performance Chip Multi Processors (CMPs). We address previously proposed CMP architectures based on Non Uniform Cache Architecture (NUCA) over NoC, analyze basic memory transactions and translate them into a set of network transactions. We first show how a simple, generic NoC which is equipped with needed module interface functionalities can provide infrastructure for the coherent access of both static and dynamic NUCA. Then we show how several low cost mechanisms incorporated into such a Vanilla NoC can facilitate CMP and boost performance of a cache coherent NUCA CMP. The basic mechanism is based on priority support embedded in the NoC, which differentiates between short control signals and long data messages to achieve a major reduction in cache access delay. The low cost Priority-based NoC is extremely useful for increasing performance of almost any other CMP transaction (i.e. uncached and cache-coherenet R/W, search in DNUCA, isolating low priority traffic, synchronization and mutual exclusion support). Priority-based NoC along with the discussed NoC interfaces are evaluated in detail using cycle-accurate CMP-NoC simulations across several SPLASH-2 benchmarks and static web content serving benchmarks showing substantial L2 cache access delay reduction and overall program speedup. For further system improvements, we introduce additional low cost NoC mechanisms that include: virtual invalidation rings, efficient store-and-forward multicast for short messages which is embedded within a wormhole NoC, and a line-cache search mechanism for the efficient operation of dynamic NUCA. These mechanisms can also expedite not only cache coherency transactions but also other basic CMP transactions such as search and serialization primitives support.

1. Introduction

Microprocessor architecture is in transition towards multi-core architectures that exploit thread-level parallelism, and provide performance improvements as well as power-efficiency. Such chip multi-processors (CMPs)[1-7] need to employ large shared on-chip cache memory (typically a L2 cache). The cache must support parallel transactions with multiple cores. Hence, a distributed cache, comprised of multiple memory banks interconnected by a network on chip (NoC)[9,31-43], as illustrated in Figure 1, is an accepted and likely approach. In such a structure, the effective access time to the shared cache will become a major performance bottleneck, as both the number of cores and the number of clock cycles required for signal propagation across the die will increase with technology scaling.

This architecture raises many challenges because the system depicted in Figure 1 needs to efficiently discover the cached location of each physical memory address and maintain multiple data copies, while ensuring data coherency of shared data among all the cores. Traditional snooping protocols for cache coherency [25] are not suitable for implementation over a NoC, and are not scalable with the number of cores. Directory-based coherence protocols require multiple network traversals (e.g. to search the cached location, determine the sharing status, update or invalidate etc.). Consequently, a CMP equipped with a standard NoC and standard processor and cache network interfaces may incur large delays in cache transactions.

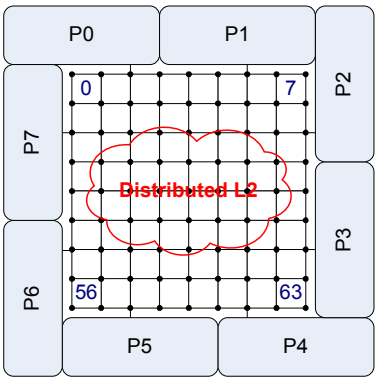

Figure 1. Modern CMP System interconnected by NoC :8 CPUs along with L2 Cache distributed in 64 Banks

Previous CMP research mainly addressed the principal architectural issues of distributed shared CMP cache over a NoC abstraction [1]-[4][6][7]. In the evaluation of SNUCA and DNUCA [1][2] the authors make simplifying assumptions regarding network delays and behavior. They do not evaluate any detailed NoC design or optimize the NoC for supporting typical cache operations.

There has been substantial prior work in the area of cache coherency optimization in the context of multiprocessors [11-18]. The majority focused on in-protocol optimizations, releasing consistency model and speculation [10][11][12][13]. Some approaches combined snooping and directory-based protocols [12]. Several studies looked into broadcast and multicast snooping, and ring optimizations [14][15][16][17]. In [18] the authors tried to efficiently map a coherency protocol onto physical wires in several metal layers, with different widths and thicknesses. The token coherence method [19] suggests to exchange and count tokens to control coherence permissions.