RETROSPECTIVE: Interconnections in Multi-Core Architectures: Understanding Mechanisms, Overheads, and Scaling

Rakesh Kumar
University of Illinois, Urbana-Champaign
Email: rakeshk@illinois.edu

Dean M. Tullsen
University of California, San Diego
Email: tullsen@ucsd.edu

I. Background

By summer of 2004, commercial multi-core processors had just started appearing in the market. These were dual-core processors designed to improve the throughput of processors. There was considerable excitement as it was apparent to many that most future processors were going to be multi-core. It was also clear that the number of cores on a multi-core processor would increase - some were predicting tens of processors on the same chip. Putting a large number of cores on the same chip presented a wide variety of problems and opportunities. What should each core look like? How should the memory hierarchy of a multi-core processor be organized? How should the cores be connected? How should the cores interact with each other to support coherence and consistency at low cost? How should one build software and hardware to ensure high utilization of the cores? A flurry of work had started both in industry and academia researching different aspects of multi-core processor architecture, design, and programming.

At that time, we had been looking at heterogeneous multi-core architectures and conjoined-core architectures as ways to improve multi-core processor efficiency. The work on heterogeneous multi-cores was primarily asking the question: should the cores on a multi-core processor be homogeneous or heterogeneous given that there exists considerable diversity among workloads? Also, what should be the architecture of each core to maximize efficiency. The work on conjoined-core architectures was asking the question: what is the right degree of sharing across the different cores of a multi-core processor. For these works, issues surrounding interconnections between cores were largely ignored. Victor Zyuban had been working on IBM's next generation server processor - POWER6 - especially in the context of power modeling and optimization. He had previously done a considerable amount of work on microarchitectural and circuit techniques for power optimization of single-core processors.

Rakesh was invited to IBM TJ Watson Research Center as a research intern for the summer of 2004. Victor was assigned as his mentor. While it was clear that Rakesh would work on some research project related to multi-core processors, it took a few days of discussion after Rakesh reached Yorktown Heights to decide that he would work on interconnections for multi-core architectures. This was a component of a multi-core processor that neither Rakesh, Victor, or Dean had looked at before.

At that time, there was limited understanding, at least in the public domain, of the design space of the interconnection framework for multi-core architectures, especially how it interacts with the rest of the architecture. The cost of implementing the interconnect for multi-cores was also unclear, especially for processors with a large number of cores – commercial multi-core processors then had only two cores while the research projects that studied larger number of cores (Stanford Hydra CMP studied a four core processor, DEC/WRL Piranha was an eight core processor) did not have silicon prototypes. The value of co-design of the interconnection network and the rest of the chip was not known. We decided to perform one of the earliest studies examining interconnect design issues and costs for multi-core processors.

II. The Study

Such a study required a deep understanding of how cores were interconnected in a multi-core processor and the associated overheads. Rakesh and Victor spent a lot of time understanding the details of implementation of interconnection on POWER4 and POWER5 – two of IBM's multi-core server processors. They also spent time understanding interconnects for chips, multi-chip modules, and board-level nodes on different IBM systems.

It became clear quickly that connecting cores on the same chip was a new challenge. Before multi-core, no one had to think about cores/caches and interconnect as a zero-sum game, because they happened on different silicon with different budgets and you could optimize them independently. But on a single chip, it is a zero-sum game, and we suspected that when you treat it as such, you would end up with some hard decisions (and some interesting tradeoffs).

Based on the above understanding, we set out to study the different interconnection frameworks for multi-core processors. All commercial multi-core processors back then were shared-memory processors. So, we decided to restrict our study to such. IBM's multi-core processors were weakly consistent. We made the same assumption about consistency. For coherence, we assumed a MESI-like snoopy write invalidate protocol. We considered bus-based and crossbar-based

interconnections. There had been some proposals by then for packet-based on-chip interconnection networks. However, both because the commercial processors then were based on buses, switches, or crossbars and because we deemed the cost of communication to be high for packet-based interconnection networks for small to moderate number of cores (which was our focus), we did not consider packet-based interconnection networks for our study. We also considered hierarchical interconnections where a point-to-point link connects two shared buses in a system with multiple shared buses.

We created area, power, and latency models for the different interconnection mechanisms and topologies. The detail and accuracy of the models went well beyond prior published work in this area at the time. These models were parameterized and allowed direct exploration of the various tradeoffs between performance and power, and between performance and area. The models also allowed a study of the sensitivity to technology, pipeline depth, number of cores, and on-chip memory sizes. The models were driven by representative commercial workloads.

We used the models to explore the design space of interconnect architectures, including crossbars, point-to-point connections, bus architectures, and various combinations of those technologies at different widths. We also explored a hierarchical bus structure that reduces local communication overheads, at the expense of cross-chip latencies.

The study delivered several strong messages to processor architects. First, the interconnect in a multi-core is a first-class component - it is performance critical and has high costs. On an 8-core processor, for example, the interconnect consumed the power equivalent of one core, took the area equivalent of three cores, and added delay that accounted for over half the L2 access latency even under conservative assumptions. Second, cores/caches and interconnect indeed are a zero sum game and co-design of cores, caches and interconnect is a must to optimize chip efficiency. For example, we showed that neither the core/cache architectures nor the interconnect architecture can be derived independently, but that the best chip design is a result of careful and hard tradeoffs between each of these elements. In fact, we showed repeatedly that design decisions made ignoring the impact of the interconnect are often the opposite of the decision indicated when these factors are properly accounted for. Third, hierarchical interconnects can mitigate high overheads of interconnection. A hierarchical approach to interconnects can exploit shorter buses with shorter latencies when traffic remains local. To increase the effectiveness of such an approach, we also studied "thread bias" - the probability that a miss is serviced on a local cache (a cache connected to the same interconnect), rather than a cache on a remote interconnect. A workload with high thread bias means that we can identify and map "clusters" of threads that principally communicate with each other on the same interconnect. We showed that a hierarchical interconnects works better than single-level interconnects even for small amounts of thread bias.

III. Looking back and forward

Being the first work to examine area and power costs and design issues for on-chip interconnects in a cache-coherent multi-core processor, it influenced the chip design and implementation community in several ways. First, it was valued by the community as an expository paper detailing issues in interconnection implementation and analysis. The understanding helped spawn a large number of works on interconnect modeling, analysis, and optimization. Second, its message that the design choices for the interconnect have significant effect on the rest of the multi-core architecture and that cores, caches, and interconnect must be co-designed to optimize efficiency resonated with many. The co-design principle was used by a large body of follow-on work on multi-core and interconnect design and implementation leading to considerable impact on research and practice. Third, many found value in the models we created to study interconnections. Many modeling assumptions we made continue to be used today. The paper was recognized for its impact with the 2020 TCCA/SIGARCH ISCA Influential Paper Award.

Going forward, we think that the paper will retain its relevance and continue to have an impact. First, the number of cores on processors has kept increasing. Today's server processors have reached 64 cores on the chip. This makes a careful design of interconnect more critical than ever. Second, new classes of processors are becoming multi-core. For more than a decade after the paper was written, microcontrollers continued to be single-core; but even they have become multi-core now. These processors have much more stringent constraints in terms of cost, area, and power. As such, interconnect optimization would be important. Third, new computing applications are emerging that use multi-core processors. Consider sensors and wearables, for example. Many sensors and wearables are driven by multi-core processors. Similarly, a multi-core approach has been shown to minimize energy for some near-threshold computing applications. Many edge analytics appliances consist of a large number of wimpy cores. For these applications, again, interconnect overheads must be minimized. Finally, it is important to note that the cost of communication continues to increase relative to the cost of computation with technology scaling. Thus, the importance of optimizing interconnect keeps increasing even for a given architecture as technology scales. Unsurprisingly, we are seeing a plethora of proposals on interconnection-aware architectures, including several in-memory and near-memory architectures, spatial architectures, waferscale architectures, and packaging-aware architectures.

Overall, as the paper advocates, understanding interconnection mechanisms, overheads, and scaling will be critical.

IV. Acknowledgement

Victor Zyuban could not participate in putting together this retrospective due to company policy. We, of course, were thrilled to have the opportunity to work with Victor on the original paper and would like to acknowledge again his leadership and contributions on this project.