RETROSPECTIVE: Scale-Out Processors

Pejman Lotfi-Kamran[1], Boris Grot[2], Michael Ferdman[3], Stavros Volos[4], Onur Koçberber[5], Javier Picorel[6], Almutaz Adileh[6], Djordje Jevdjic[7], Sachin Idgunji[8], Emre Ozer[9], and Babak Falsafi[10]

[1]IPM [2]University of Edinburgh [3]Stony Brook University [4]Microsoft [5]Oracle
[6]Huawei [7]National University of Singapore [8]NVIDIA [9]Pragmatic Semiconductor [10]EPFL

I. Context

In 2012, datacenters were growing at phenomenal speed, with forecasts of unprecedented levels of electricity consumption and emissions reaching that of the airline industry. These forecasts were exacerbated by the slowdown in Dennard Scaling and extraordinary projections for chip power density. Unfortunately, datacenters were built with volume servers that inherited the basic hardware and OS organization of the 90s' desktop PCs, with server cost (and not silicon efficiency) as the key design criterion. Many had established that there is a fundamental mismatch between desktop CPU microarchitecture and the silicon requirements of scale-out datacenter services.

At that time, novel server platforms emerged with energy-efficient ARM (e.g., Calxeda, Marvell, SeaMicro) and MIPS (e.g., Tilera) cores. Because memory played a pivotal role in the cost and the basic organization of emerging scale-out services in datacenters, 32-bit ARM cores were ill-suited in the server setting and never found traction. Similarly, manycore tiled CPUs were mostly optimized for on-chip communication in parallel workloads rather than supporting server software stacks that primarily benefited from request-level parallelism and exhibited little communication across threads. The Scale-Out Processors was the product of an EU-funded project, EuroCloud Server, among ARM, EPFL, IMEC, Nokia, and University of Cyprus, to design cloud-native servers with 64-bit out-of-order ARM cores (derived from Cortex A-15) and 3D-stacked DRAM running cloud services.

II. Server Workloads

The microarchitectural requirements of commercial server workloads and their mismatch with desktop CPUs were identified in several studies in the 90s [1], [6]. In the decade that followed, many also re-evaluated proper provisioning of the memory hierarchies and the trade-off between single-thread performance and throughput for database and web workloads [5]. These studies eventually led to the first generation of workload-optimized chips (e.g., Sun Niagara and DEC Piranha) for commercial servers.

In 2011, Hardavellas, et al. [4] were studying the impact of technology scaling on server architecture with commercial server workloads. Their results clearly indicated that for the combination of stringent chip power constraints, emerging high-bandwidth and energy-efficient memory fabrics, and the abundance of request-level parallelism in server workloads, a custom manycore CPU would be optimal for throughput, power, and area in servers. Unlike conventional CPU designs, such custom CPUs would have minimal on-chip memory (i.e., MBs) to hold the instruction working set of deep server software stacks, and minimal complexity (i.e., power, area) cores to access off-chip data and exploit request-level parallelism across server threads.

With the emergence of open-source server software stacks, Ferdman, et al. evaluated for the first time a suite of scale-out workloads and identified the microarchitectural mismatch between the workloads and the servers [2]. These findings were consistent with the characterization of the server workloads of the 90s. In particular, they listed several microarchitectural characteristics salient in scale-out workloads that were drastically different from desktop workloads: (1) instruction supply bottleneck due to large instruction working sets in server software stacks, (2) low instruction-level (ILP) and memory-level (MLP) parallelism in server software stacks, (3) secondary data working sets that are orders of magnitude larger than on-chip memory, and (4) low on- and off-chip per-thread memory traffic. We also later observed that the workloads do not really use either floating-point arithmetic or vector operations. These requirements were at odds with state-of-the-art volume servers at the time, which were being shipped with half a dozen ILP-centric desktop x86 cores padded with 12MB of LLC per socket, with two sockets serving 10s of GB of DRAM.

III. Scale-Out Processors

To maximize silicon efficiency in a cloud-native processor, we understood that the chip resources should be properly provisioned to maximize throughput. But more importantly, we hypothesized that a conventional "scale-up" processor organization may not be sufficient or necessary for maximum efficiency. Cache coherence requires prohibitive amounts of silicon with a growing number of cores. Moreover, scaling the scope of shared hardware resources (e.g., core count, memory hierarchy capacity) in scale-out workloads has diminishing returns in load balancing and increasing throughput [9].

We proposed *Scale-Out Processors* [8] that organize silicon resources into multiple physical servers called "pods". Each pod runs a full software stack and has its data sharded across physical memory partitions, but shares memory and I/O ports and pins at the chip level with other pods [3]. Because pods operate as independent servers with no contention on microarchitectural resources, optimally sizing a pod and scaling the

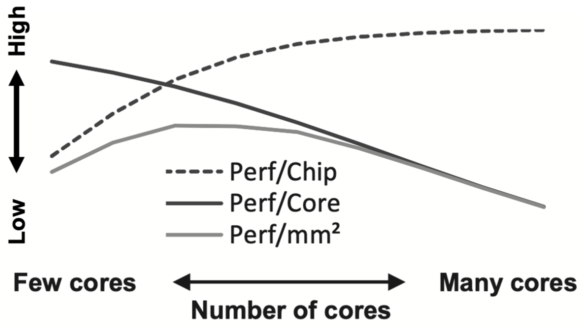

Fig. 1. Identifying the sweet spot for throughput per area.

number of pods per chip would allow for a linear increase in throughput while maintaining the proper server CPU chip core-to-cache ratio and optimizing overall datacenter costs at the board and system level [3].

Scale-Out Processors introduced a unified metric, named performance density (PD), to measure silicon efficiency as the performance delivered per square millimeter of the die [8]. PD provides a straightforward method to contrast different designs that use the same core microarchitecture, but vary in aspects such as core count, last-level cache (LLC) size, and interconnect parameters.

Figure 1 provides an intuitive presentation of the PD concept using a hypothetical server workload. The x-axis represents the core count (where each core has private L1 I/D caches) and a fixed-size LLC (i.e., L2) shared among them. Moving right on the graph, the pod area and core-to-cache ratio increase. The solid black line illustrates per-core throughput, which decreases as the core count increases due to the higher distance between cores and the LLC, slowing down the instruction supply. The dashed line depicts the total throughput, which increases with more cores, but levels out due to lower per-core throughput. Finally, the gray line corresponds to PD which peaks at the optimal point that strikes a balance between core area, core count, LLC size, and the distance to the LLC. For any fixed-size LLC (e.g., 1-8MB [4]), the sweet spot moves to the right (not shown) with leaner cores requiring less area per core. Likewise, a larger LLC (not shown) moves the sweet spot to more cores per pod to amortize the LLC area.

Our key findings were:

- Conventional server chips (e.g., Intel's) were an order of magnitude lower in efficiency because of: (1) big ILP-centric cores with unused silicon in server workloads, and (2) a large but slow LLC with mostly unused capacity creating an instruction supply bottleneck [5].
- Tiled server chips (e.g., Tilera's) had more efficient cores, but wasted significant silicon for the LLC per tile. Their on-chip interconnect, which optimized for parallel workloads with core-to-core traffic, was a mismatch for pods that mostly exhibited core-to-LLC traffic [7]. The net result was a much lower core-to-cache ratio in silicon area and 2x lower efficiency.

IV. The Paper's Legacy

This body of work [2], [8] laid the foundation for an ARM-based cloud-native CPU, Cavium ThunderX, which employed 48 in-order ARM cores with 78K of L1I with 16MB (0.3MB per core) of shared L2 (as LLC), an order-of-magnitude larger core-to-cache silicon ratio than conventional server CPUs.

A lasting impact of the work has been to accelerate instruction supply. x86 and ARM cores alike now boast 4K-6K BTB entries to allow the front end to fetch ahead. While Intel and AMD have also continued the trend of building large LLCs (1.5-4MB per core) which are overprovisoned for scale-out workloads, both have transitioned to 2MB private L2 caches for fast instruction supply. The Ampere cloud-native CPU, AmpereOne, also uses 2MB private L2s, but with only 0.3MB of LLC per core for superior silicon efficiency.

PD remains a first-order metric to evaluate silicon efficiency, especially with the emergence of the post-Moore era and heterogeneous logic, with area serving as a proxy for dynamic power. With microarchitectural characteristics of monoliths and microservices continuing to exhibit both front-end and back-end bottlenecks in CPUs, PD can shed light on how much silicon to provision in cores, cache hierarchies, accelerators, network, I/O, and glue logic to maximize throughput while maintaining latency guarantees in datacenter services. With 3-4x smaller cores, out-of-order ARM cores today (e.g., Ampere) achieve a high overall efficiency even at half of the average per-core throughput for scale-out workloads. The addition of SMT [2] may be a promising approach to further increase silicon efficiency in cloud-native CPUs.

References

[1] L. A. Barroso, K. Gharachorloo, and E. Bugnion, "Memory system characterization of commercial workloads," in *Proceedings of the 25th Annual International Symposium on Computer Architecture*, 1998.

[2] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Quantifying the mismatch between emerging scale-out applications and modern processors," *ACM Transactions on Computer Systems*, vol. 30, no. 4, 2012.

[3] B. Grot, D. Hardy, P. Lotfi-Kamran, B. Falsafi, C. Nicopoulos, and Y. Sazeides, "Optimizing data-center tco with scale-out processors," *IEEE Micro*, vol. 32, no. 5, 2012.

[4] N. Hardavellas, M. Ferdman, B. Falsafi, and A. Ailamaki, "Toward dark silicon in servers," *IEEE Micro*, vol. 31, no. 4, 2011.

[5] N. Hardavellas, I. Pandis, R. Johnson, N. Mancheril, A. Ailamaki, and B. Falsafi, "Database servers on chip multiprocessors: Limitations and opportunities," in *Proceedings of the Third Biennial Conference on Innovative Data Systems Research, 2007.*

[6] K. Keeton, D. A. Patterson, Y. Q. He, R. C. Raphael, and W. E. Baker, "Performance characterization of a quad pentium pro SMP using OLTP workloads," in *Proceedings of the 25th Annual International Symposium on Computer Architecture*, 1998.

[7] P. Lotfi-Kamran, B. Grot, and B. Falsafi, "Noc-Out: Microarchitecting a Scale-Out Processor," in *45th Annual IEEE/ACM International Symposium on Microarchitecture, 2012.*

[8] P. Lotfi-Kamran, B. Grot, M. Ferdman, S. Volos, O. Kocberber, J. Picorel, A. Adileh, D. Jevdjic, S. Idgunji, E. Ozer, and B. Falsafi, "Scale-Out Processors," in *Proceedings of the 39th International Symposium on Computer Architecture*, 2012.

[9] S. Novakovic, A. Daglis, E. Bugnion, B. Falsafi, and B. Grot, "The case for rackout: Scalable data serving using rack-scale systems," in *Proceedings of the Seventh ACM Symposium on Cloud Computing, 2016.*