RETROSPECTIVE: Memory System Characterization of Commercial Workloads

Luiz André Barroso
Google
luiz@google.com

Kourosh Gharachorloo
Google
kourosh@google.com

Edouard Bugnion
EPFL, Switzerland
edouard.bugnion@epfl.ch

***Abstract*—The paper was published at ISCA 1998, exactly 25 years ago [3]. At the time, the community relied primarily on trace-driven, user-level simulation of scientific workloads to study architectural tradeoffs in processor design. This paper looked instead at commercial workloads, which by then had already surpassed scientific workloads in terms of market share for high-end servers. It also broke new ground by using two distinct methodologies that were emerging at the time: (1) tools to capture low-overhead hardware performance counters and (2) complete system simulation that could selectively tradeoff simulation performance for simulation accuracy. The combination was used to study the cache hierarchy and shared-memory communication patterns on performance.**

Today the tools, methods, and workloads we used are broadly available to the community. While the hardware environment has obviously changed significantly, the need to scientifically study CPU pipelines, intra-socket communication, and board-level communication in detail has not. For example, the current architectural disruption caused by machine learning acceleration provides a good opportunity for the community to use similar methods.

I. The 1998 paper

The paper focused on commercially-relevant workloads using state-of-the-art commercial hardware and software of the era. The target environment was a 4-socket AlphaServer with 2GB of RAM running Digital UNIX, the Oracle database server, and the Altavista Search Engine. As now, caches were coherent, inclusive, and organized in three levels, yet their size and associativity were lower than today and the L3 cache was off-chip. Our hypothesis was that memory system performance was much more critical to commercial workloads than it was for the scientific benchmarks typically used in architectural studies.

We used hardware performance counters, a relative novelty at the time, and the DCPI tool [2] to study these workloads *in situ* with low perturbation. For some studies, application binaries were further instrumented using the ATOM binary translator [21]. While the Alpha processor used was a statically-scheduled, dual-issue CPU, the results showed that the performance was dominated by stalls due to instruction and data cache misses. The OLTP workload had an exceptionally poor CPI (cycles per instruction) of 7.0 largely due to instruction cache misses, while the other workloads had a CPI between 1.3 and 1.5 (ideal: 0.5). The detailed breakdown could attribute stalls due to instruction, data, L2, L3, TLB misses, and branch misprediction. Finally, it could quantify the fraction of cache misses that required cross-socket communication, which was (and remains) particularly expensive.

We then used the SimOS complete system simulator [18] to analyze the sensitivity to key aspects of the cache hierarchy, in particular size and associativity of caches. SimOS was one of the first-generation complete system simulators capable of emulating a multiprocessor server and its I/O devices with enough accuracy to run unmodified operating systems. These tools must —back then as well as today— provide a way to switch between multiple simulation modes that trade-off workload performance and accuracy of the simulation differently. For this paper, we added SimOS support to Alpha processors, including a fast mode using dynamic binary translation (similar to Embra for MIPS [23]) and a detailed model that used a conventional interpreter connected to a detailed memory model.

We could therefore run the same workloads with two totally different methodologies, *i.e.,* low-perturbation hardware counters and complete simulation. This paper was one of the first papers to validate the correspondence of results; for example, instruction counts differed by ca 1% between the two setups.

Yet the primary use of the simulation was to change the cache hierarchy. We could observe then, as is now well-known, that 2-way set associative caches performed as well as direct-mapped caches with twice the capacity. We used SimOS's cache miss classification engine, based on the theory of Dubois *et al.* to separate communication misses from conflict and capacity misses, and to further split communication misses between true and false sharing [9]. This provided insights into the locality and communication patterns of these workloads, which could be measured separately for the application portion (user level) and the operating system execution of each workload.

II. Impact

The paper demonstrated to the architecture community that complex computer systems running commercial workloads could be analyzed through instrumentation and simulation.

The results of the paper are used as the primary case study in the chapter on the performance of symmetric shared-memory multiprocessors in Hennessy & Patterson's textbook (chapter 5.3 of the 5th edition) [13].

Beyond the computer architecture community, the approach was used contemporarily to study operating systems [19] and databases [1], [16]. Together, these studies were instrumental in bringing to the attention of the community the importance of commercial workload and the key microarchitectural differences with scientific workloads and benchmarks. These insights led a team at DEC WRL to design Piranha, one of the first scalable multicore processors with a cache hierarchy specifically designed with commercial workloads in mind [4].

These workloads proved challenging across time: the cycles-per-instruction metrics for OLTP and decision support workloads barely improved by a factor of 2× over the subsequent decade after our paper was published [12], and have not substantially changed since.

The combined methodology of hardware-based profiling and complete system simulation is now mainstream. Hardware counters are now ubiquitous and considered necessary for any serious workload tuning, with tools such as perf and dtrace [8]. Complete machine simulation is now considered table-stakes in our community. SimOS was limited to MIPS and Alpha architectures and deprecated relatively quickly. SimICS [17], SimFlex [22], QEMU [6], GEM5 [7], and many others are all considered essential parts of the computer architecture toolbox.

The same methodology used to characterize memory system performance was employed more recently with the characterization of the CloudSuite benchmark, which showed that instruction caches remain a key performance bottleneck [10]. Google authors made similar observations when profiling a warehouse-scale computer [15] and understanding software dynamics [20].

III. Outlook

The tools and methodologies of this paper remain valid today. More than ever, the field of computer architecture is still concerned with optimizing hardware for particular workloads and for particular system software assumptions such as virtualization, or more recently confidential computing. While complete system simulation (*e.g.,* GEM5) has become the workhorse used by many computer architects, its performance limitations when running detailed models still hover at around 250 KIPS, which severely limits the scope of applicability of studies.

While current web workloads have similar memory system bottlenecks as the prior generation of scale-up commercial workloads, they introduce microsecond-scale interactions between servers which are the primary cause of datacenter tax of today's workloads [5]. The current shift towards ML/AI workloads and use of specialized hardware accelerators for both computation [14] and data streaming [11] poses a new class of challenges to architects, and a new class of opportunities for complete system simulators.

Acknowledgements

The original work was performed at Digital Equipment Corporation's Western Research Laboratory, with key support from the DEC database group, the Stanford SimOS team, and the AltaVista team (especially Mike Burrows). We are honored to be included in this retrospective of papers that celebrates ISCA's 50th anniversary.

References

[1] A. Ailamaki, D. J. DeWitt, M. D. Hill, and D. A. Wood, "DBMSs on a Modern Processor: Where Does Time Go?" in *VLDB*, 1999, pp. 266–277.

[2] J.-A. M. Anderson, L. M. Berc, J. Dean, S. Ghemawat, M. R. Henzinger, S.-T. Leung, R. L. Sites, M. T. Vandevoorde, C. A. Waldspurger, and W. E. Weihl, "Continuous Profiling: Where Have All the Cycles Gone?" in *SOSP*, 1997, pp. 1–14.

[3] L. A. Barroso, K. Gharachorloo, and E. Bugnion, "Memory System Characterization of Commercial Workloads." in *ISCA*, 1998, pp. 3–14.

[4] L. A. Barroso, K. Gharachorloo, R. McNamara, A. Nowatzyk, S. Qadeer, B. Sano, S. Smith, R. Stets, and B. Verghese, "Piranha: a scalable architecture based on single-chip multiprocessing." in *ISCA*, 2000, pp. 282–293.

[5] L. A. Barroso, M. Marty, D. A. Patterson, and P. Ranganathan, "Attack of the killer microseconds." *Commun. ACM*, vol. 60, no. 4, pp. 48–54, 2017.

[6] F. Bellard, "QEMU, a Fast and Portable Dynamic Translator." in *USENIX Annual Technical Conference, FREENIX Track*, 2005, pp. 41–46.

[7] N. L. Binkert, B. M. Beckmann, G. Black, S. K. Reinhardt, A. G. Saidi, A. Basu, J. Hestness, D. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. S. B. Altaf, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator." *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.

[8] B. Cantrill, M. W. Shapiro, and A. H. Leventhal, "Dynamic Instrumentation of Production Systems." in *USENIX Annual Technical Conference*, 2004, pp. 15–28.

[9] M. Dubois, J. Skeppstedt, L. Ricciulli, K. Ramamurthy, and P. Stenström, "The Detection and Elimination of Useless Misses in Multiprocessors." in *ISCA*, 1993, pp. 88–97.

[10] M. Ferdman, A. Adileh, Y. O. Koçberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Clearing the clouds: a study of emerging scale-out workloads on modern hardware." in *ASPLOS-XVII*, 2012, pp. 37–48.

[11] D. Foley and J. Danskin, "Ultra-Performance Pascal GPU and NVLink Interconnect." *IEEE Micro*, vol. 37, no. 2, pp. 7–17, 2017.

[12] N. Hardavellas, I. Pandis, R. Johnson, N. Mancheril, A. Ailamaki, and B. Falsafi, "Database Servers on Chip Multiprocessors: Limitations and Opportunities." in *CIDR*, 2007, pp. 79–87.

[13] J. L. Hennessy and D. A. Patterson, *Computer Architecture - A Quantitative Approach, 5th Edition.* Morgan Kaufmann, 2012.

[14] N. P. Jouppi *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit." in *ISCA*, 2017, pp. 1–12.

[15] S. Kanev, J. P. Darago, K. M. Hazelwood, P. Ranganathan, T. Moseley, G.-Y. Wei, and D. M. Brooks, "Profiling a warehouse-scale computer." in *ISCA*, 2015, pp. 158–169.

[16] 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 *ISCA*, 1998, pp. 15–26.

[17] P. S. Magnusson, M. Christensson, J. Eskilson, D. Forsgren, G. Hållberg, J. Högberg, F. Larsson, A. Moestedt, and B. Werner, "Simics: A Full System Simulation Platform." *Computer*, vol. 35, no. 2, pp. 50–58, 2002.

[18] M. Rosenblum, E. Bugnion, S. Devine, and S. A. Herrod, "Using the SimOS Machine Simulator to Study Complex Computer Systems." *ACM Trans. Model. Comput. Simul.*, vol. 7, no. 1, pp. 78–103, 1997.

[19] M. Rosenblum, E. Bugnion, S. A. Herrod, E. Witchel, and A. Gupta, "The Impact of Architectural Trends on Operating System Performance." in *SOSP*, 1995, pp. 285–298.

[20] R. L. Sites, *Understanding Software Dynamics.* Addison-Wesley Professional Computing Series, 2021.

[21] A. Srivastava and A. Eustace, "ATOM - A System for Building Customized Program Analysis Tools." in *PLDI*, 1994, pp. 196–205.

[22] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe, "SimFlex: Statistical Sampling of Computer System Simulation." *IEEE Micro*, vol. 26, no. 4, pp. 18–31, 2006.

[23] E. Witchel and M. Rosenblum, "Embra: Fast and Flexible Machine Simulation." in *SIGMETRICS*, 1996, pp. 68–79.