Criticality Aware LLC Partitioning: Reducing System Turnaround Time with Intel CAT

Lucia Pons, Vicent Selfa, Julio Sahuquillo, Salvador Petit, Julio Pons
Dept. of Computer Engineering
Universitat Politècnica de València, Spain
{lupones,viselol,jsahuqui,spetit,jpons}@disca.upv.es

1 INTRODUCTION

Resource sharing is a major concern in current multicore processors. Among the shared system resources, the Last Level Cache (LLC) is one of the most critical, since destructive interferences between applications accessing it imply more off-chip accesses to main memory, which incur in long latencies that can severely impact the overall system performance. To help alleviate this issue, current processors implement huge LLCs, but even so, inter-application interferences can harm the performance of a subset of the running applications when executing multiprogram workloads. For this reason, recent Intel processors feature Cache Allocation Technologies (CAT) to partition the cache and assign subsets of cache ways to Classes Of Service (CLOSes), groups of applications (PIDs) or logical cores [3]. Figure 1 shows an example of a possible CAT configuration for a cache with 20 ways using 2 CLOSes, each holding 2 applications. One of the most interesting features of CAT is that it is possible to change its configuration dynamically, allocating more cache space, for example, to a CLOS containing an application that has a spike in activity and requires it.

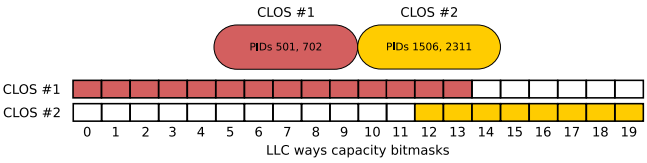

Figure 1: Intel CAT example with PIDs associated to two CLOSes.

2 CRITICAL-AWARE PARTITIONING APPROACH

This poster proposes the **Critical-Aware (CA)** LLC partitioning approach, which leverages CAT and improves the performance of multiprogram workloads. CA separates, at runtime, critical applications from non-critical. That is, it distinguishes applications whose performance is being damaged by LLC sharing from others that are unaffected. The later only need few ways to achieve their maximum performance but occupy a large fraction of the cache space. Our proposal, by design, divides the cache in two partitions (one for each type of applications), allocating a greater amount of cache space to the partition holding the critical applications. Since not all the cache critical applications show the same criticality, CA tries to further refine the partitions, dynamically readjusting their size.

CA manages to significantly improve the turnaround time of multiprogram workloads. This time is defined as the elapsed time between the workload mix execution start and the instant the last application of the workload mix finishes. Turnaround time is especially important in batch-based systems, since improving this performance metric allows the system to transit to a low power state, saving energy. Eyerman and Eeckhout [1] claim that program turnaround time in general-purpose systems and interactive environments should be one of the primary performance criteria. Figure 2 shows the turnaround times (in intervals of half a second) of both CA and the reference system across the studied mixes, containing each 8 applications with 1, 2 or 3 critical applications. The workload mixes were randomly generated using 25 applications from the SPEC CPU2006 [2] benchmark suite, taking into account that the number of critical applications is much lower than the number of non-critical ones. Experimental results show that CA improves turnaround time on average by 15%, and up to 40% compared to a baseline system without partitioning.

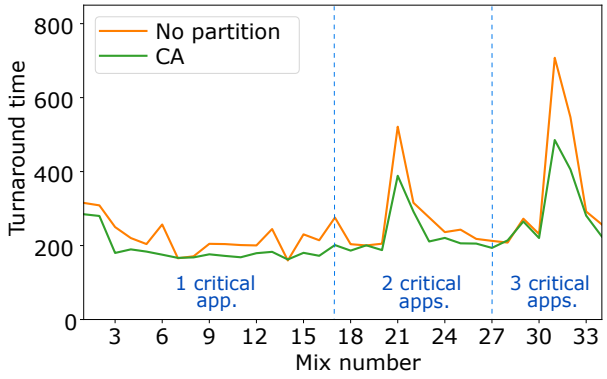

Figure 2: Turnaround time of CA vs reference system of individual workload mixes.

REFERENCES

[1] S. Eyerman and L. Eeckhout. 2008. System-Level Performance Metrics for Multiprogram Workloads. *IEEE Micro* 28, 3 (May 2008), 42–53. https://doi.org/10.1109/MM.2008.44
[2] John L. Henning. 2006. SPEC CPU2006 Benchmark Descriptions. *Comput. Archit. News* 34, 4 (Sept. 2006), 1–17. https://doi.org/10.1145/1186736.1186737
[3] A. Herdrich, E. Verplanke, P. Autee, R. Illikkal, C. Gianos, R. Singhal, and R. Iyer. 2016. Cache QoS: From concept to reality in the IntelÂő XeonÂő processor E5-2600 v3 product family. In *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Vol. 00. 657–668. https://doi.org/10.1109/HPCA.2016.7446102