C^2AFE: Capacity Curve Annotation and Feature Extraction for Shared Cache Analysis

Cesar Gomes and Mark Hempstead
Tufts University
Medford, MA
Email: cesar.gomes@tufts.edu, mark.hempstead@tufts.edu

***Abstract*—Recent years see cache partitioning techniques employ commercial way partitioning to great effect. Techniques like partitioning build analysis on miss curves. However, the various forms of partitioning run into the issue of core scaling, requiring multi-tenant partitions and analysis of how workloads behave when sharing ever shrinking resources. We observe surprising variations in performance which can be disjoint with variations in cache misses that encourage such analysis. We present Capacity Curve Annotation and Feature Extraction, methodologies for annotation, feature extraction, and analysis of sensitivity studies based on performance curves.**

I. Introduction

Cache characterizations study workloads in isolation, building intuition on misses as a function of cache size, or miss curves [3, 5, 6, 9]. However, miss curves can misinform decisions for workloads that share cache, especially in light processors featuring way partitioning [2]. Figure 1(a) shows isolated curves (*Iso*) for a 6 million instruction trace of 519.lbm from the SPEC 2017 suite [8]. We warm cache up with the first 1 million instructions and simulate the rest to gather statistics. X-axis is cache size in the set S*. Y-axis is IPC and misses respective of top and bottom plots. We compare *Iso* (black) to shared curves (*Shr*) chosen for small, medium, and large differences between the area under each miss curve and the *Iso* case (e.g. miss curve$_{\text{Iso}}$-miss curve$_{\text{shr}}$). Gray lines mark where performance begins increasing (top), and where the *Iso* miss curve indicates a decent cache size (bottom). *Iso* indicates a smaller size than what 519.lbm_r requires in cache. It is clear that *Iso* miss curves do not always predict performance in either isolated or shared context.

Figure 1(b) shows workload behavior classes (discussed in II) and changes to behavior when LLC is shared. We see consistency in behavior for one workload (500.perlbench) in *Shr*, but others change in ways not always captured by *Iso* miss curves. This work summarizes **Capacity Curve (C^2) Annotation and Feature Extraction**, our analysis of feature and curve behavior alignment to describe feature and behavioral consistency of C^2s. We use a modified version of ChampSim [4], which models Intel Skylake [1]. We study how sharing LLC impacts pairs of 82 traces from the Problem-based Benchmark Suite (PBBS), SPEC 2006, and SPEC 2017 [7, 8]. We pair each workload with all other workloads, and

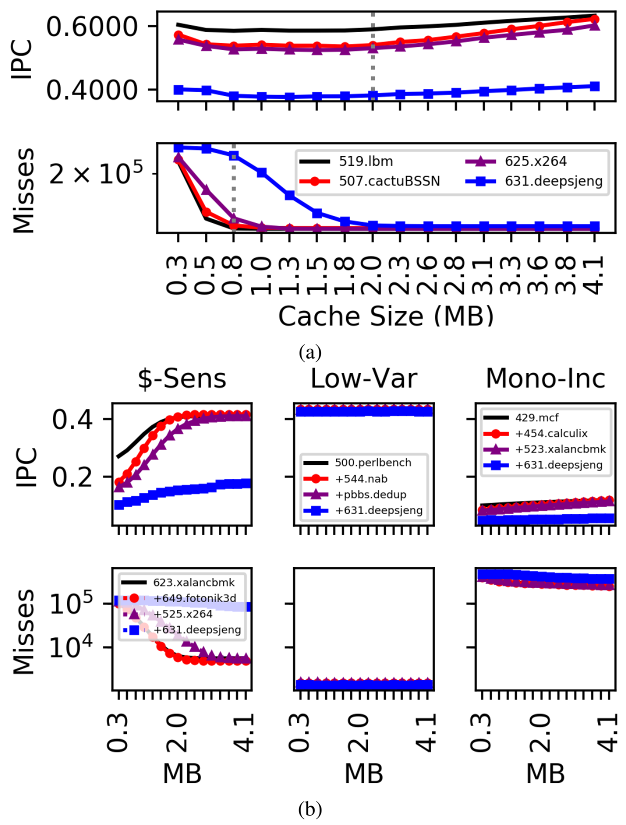


Fig. 1. Capacity Curves: (a) 519.lbm IPC and Miss C^2 demonstrate misses are not predictive in isolation, but C^2s gathered in shared context can offer better insight; (b) examples C^2 behavior classes and the changes incurred between curves of a given workload gathered in *Iso* and *Shr* context.

simulate them on each cache size in our sweep range. Total simulations number 108,896 to create 6806 C^2s.

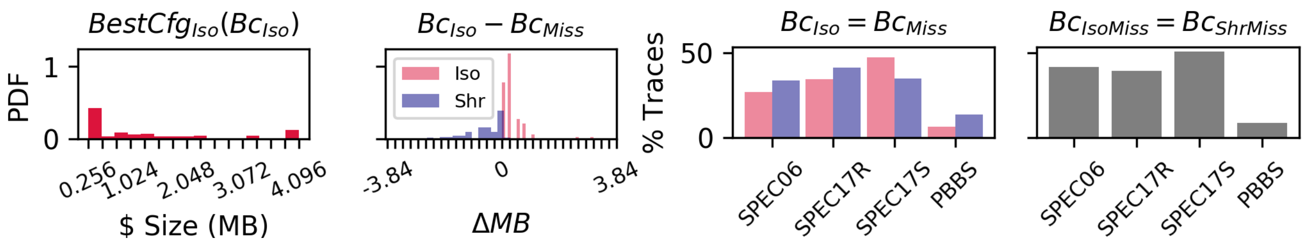
II. Capacity Curves

Capacity Curves (C^2) are created by plotting metrics (IPC, Misses, etc) against cache size within a range set by cache associativity. In this section, we summarize C^2 annotations and study features, and behavioral classes defined in Table I.

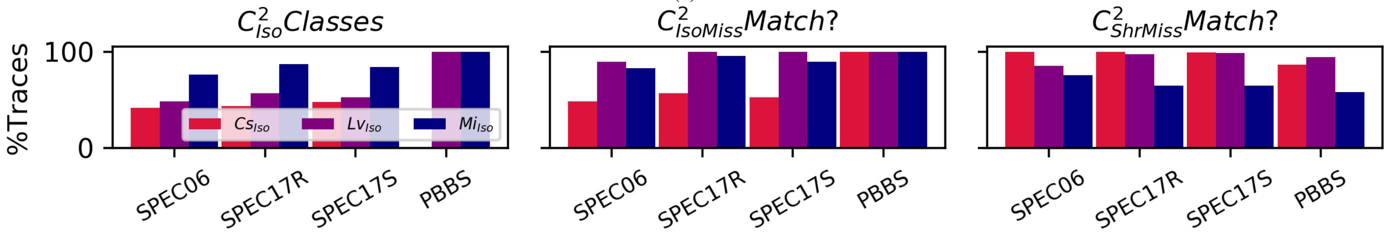
a) Curve Annotation: C^2s can express where performance is best (best config, Bc†), where metrics transition to or from steady state (knee), and how knees differ. Figure 1(a) shows 4MB as Bc for 519.libquantum. *Iso* MPKI C^2

*S is the set of cache sizes ranging from 0.256MB to 4.096MB (i.e. 1 way=0.256MB, 2 ways=0.512MB,...,16 ways=4.096MB)

†Best Cache Config (Bc) is the cache configuration that offers the best performance for a workload when not sharing LLC.



(a)



(b)

Fig. 2. Feature & Class Change Distributions: (a) In order: Distribution of Best Cache Configuration from isolated simulations ($\text{Bc}_{\text{Iso Perf}}$); Comparison of \$ size difference between $\text{Bc}_{\text{Iso Perf}}$ & Bc_{Miss} chosen based on misses gathered in *Iso* and *Shr* context (left of 0 indicates over-provision, right indicates under-provision); Comparison across each benchmark suite of % of traces where $\text{Bc}_{\text{Iso Perf}}$ aligns with $\text{Bc}_{\text{Iso Miss}}$s and $\text{Bc}_{\text{Shr Miss}}$s; Summary of % of traces where $\text{Bc}_{\text{Iso Miss}}$s and $\text{Bc}_{\text{Shr Miss}}$s match. (b) In order: Percent trace C^2_{Iso} behavior classes per suite; Percent of behavior agreement between C^2_{Iso} and $\text{C}^2_{\text{Iso Miss}}$ per suite; Percent of behavior agreement between C^2_{Iso} and $\text{C}^2_{\text{Shr Miss}}$ per suite.

TABLE I
FEATURES & CLASSES

FEATURE	**DEFINITION**
IPC Knee (Ik)	$\Delta IPC_{r-s_{Ik}}/IPC_s < 0.01$‡ $r < s_{Ik}, \forall r, s_{Ik} \in S*$
MPKI Knee (Mk)	$\Delta MPKI_{s_{Mk}-r}/MPKI_N < 0.01$ $r < s_{Mk}, \forall r, s_{Mk} \in S$
Knee Difference (Kd)	$s_{Ik} - s_{Mk}$
CLASS	**DEFINITION**
Cache Sensitive (Cs)	$\Delta IPC_{4MB-0.2MB}/IPC_{0.2MB} > 0.01$
Low Variation (Lv)	$\Delta IPC_{Bc-s}/IPC_{Bc} < 0.01, \forall s \in S$
Monotonic Increase (Mi)	$IPC_r \leq IPC_s, \forall r < s, r, s \in S$

transitions from 40 to 25 (-37%) between 0.256 and 0.768MB and settles after with no further changes, what we call a knee. The sudden dropping, leveling, and a gentle increase in IPC between 0.768 and 1.024MB is not captured in MPKI. The miss knee (Mk) in *Shr* MPKI C^2 where lbm shares LLC with 631.deepsjeng aligns with the start of IPC increase. Indeed, there exists *Shr* C^2s that better describe performance dependence given enough contention. We discuss the features we derive from these annotations next.

b) Curve Feature Distribution: In figure 2(a), we analyze over- and under-provisioning of resources when basing configuration selection on *Iso* and *Shr* miss C^2s. From the left, we see *Iso* C^2s have Bc_{Miss}s at small cache sizes in the first plot. The second plot shows miss knees (Mk) highlights over- or under-provisioning LLC resources with high frequency [‡]. The third plot shows the percent of traces have a Bc_{Miss} that matches Bc_{Iso}, where we see $\text{Bc}_{\text{Shr Miss}}$ more frequently aligns to $\text{Bc}_{\text{Iso IPC}}$ than $\text{Bc}_{\text{Iso Miss}}$. The fourth plot shows the percent of traces where $\text{Bc}_{\text{Iso Miss}}$ and $\text{Bc}_{\text{Shr Miss}}$ match, suggesting there exists feature resilience to higher contention. The next section discusses C^2 behavior classes.

c) Curve Behavior Classification: Figure1(b) shows workloads which we classify as cache sensitive (623.xalancbmk), low variation (500.perlbench), and increase monotonously (429.mcf) according to definitions in table I. Figure 2(b) shows the following per suite: behavioral class makeup; the percent of $\text{C}^2_{\text{Iso Miss}}$s that match C^2_{Iso} behavior; a similar analysis of $\text{C}^2_{\text{Shr Miss}}$s. The first plot shows SPEC suites (06-17R/S) are are 65% non-decreasing, 50% variable, and 40-45% cache sensitive, while PBBS traces are insensitive, have low variation, and do not decrease. On observation, PBBS traces does align with this behavior classification. The second plot shows C^2_{Iso} and $\text{C}^2_{\text{Iso Miss}}$ agree on PBBS behavior. Also shown is SPEC is 90-95% in agreement on Lv behavior and 80-85% in agreement on Mi behavior, but only agrees on 50% of Cs behavior, which highlights misalignment in performance and cache utilization in *Iso*. The third plot shows C^2_{Iso} and $\text{C}^2_{\text{Shr Miss}}$ disagree on PBBS behavior, which makes sense given PBBS largely fits within core cache and rarely relies on LLC. Also shown is SPEC Cs classification by $\text{C}^2_{\text{Shr Miss}}$ agrees with C^2_{Iso}, implying sharing LLC reveals sensitivity better. Further, we see the Lv classifications are 90-95% in agreement, similar to $\text{C}^2_{\text{Iso Miss}}$.

III. Conclusion & Future Work

We present Capacity Curve (C^2) Annotation and Feature Extraction as a method for analysis of workloads in shared caches. Miss curves are shown to be misleading in shared cache analysis, and the impact sharing LLC has on workloads is discussed. We summarize analysis on changes in C^2 features and behavior classes. Analysis of workloads under contention can offer insight and better alignment between miss curves and performance if the workload is cache sensitive. Future work includes studying novel metrics of shared behavior, investigating applications of C^2 in computer architecture, and to identify workloads which align miss behavior with performance goals.

[‡]Miss knees derived from changes in misses reducing to less than 1%; Many-core systems often operate with guarantees that limit critical performance drop to 1% of a goal, so we use 0.01 to replicate this.

References

[1] J. Doweck et al. “Inside 6th-Generation Intel Core: New Microarchitecture Code-Named Skylake”. In: *IEEE Micro* 37.2 (Mar. 2017), pp. 52–62.

[2] Andrew Herdrich et al. “Cache QoS: From concept to reality in the Intel® Xeon® processor E5-2600 v3 product family”. In: *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*. IEEE. 2016, pp. 657–668.

[3] Aamer Jaleel. “Memory Characterization of Workloads Using Instrumentation-Driven Simulation A Pin-based Memory Characterization of the SPEC CPU 2000 and SPEC CPU 2006 Benchmark Suites”. In: *VSSAD Technical Report 2007*.

[4] Jinchun Kim. *ChampSim*. https://github.com/ChampSim/ChampSim. 2017.

[5] Anurag Mukkara, Nathan Beckmann, and Daniel Sanchez. “Whirlpool: Improving dynamic cache management with static data classification”. In: *ACM SIGARCH Computer Architecture News* 44.2 (2016), pp. 113–127.

[6] A. Sembrant, D. Black-Schaffer, and E. Hagersten. “Phase behavior in serial and parallel applications”. In: *2012 IEEE International Symposium on Workload Characterization (IISWC)*. Nov. 2012, pp. 47–58.

[7] Julian Shun et al. “Brief announcement: the problem based benchmark suite”. In: *Proceedings of the twenty-fourth annual ACM symposium on Parallelism in algorithms and architectures*. ACM. 2012, pp. 68–70.

[8] Standard Performance Evaluation Corporation. *SPEC Benchmark Suite*. http://www.spec.org.

[9] Q. Zou et al. “Temporal characterization of SPEC CPU2006 workloads: Analysis and synthesis”. In: *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)*. Dec. 2012, pp. 11–20.