Supplementary : Non-Fusion Based Coherent Cache Randomization Using Cross-Domain Accesses

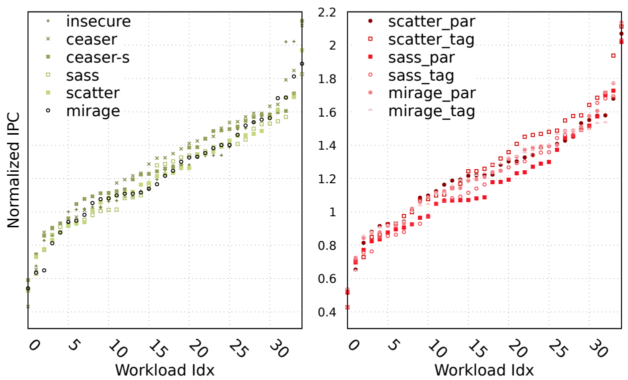

Figure 1: The IPC results for SPEC workload mixes. Each configuration has its results sorted in increasing order. Half of the configurations are presented in the left graph, and the other half in the right graph. (see Appendix §A)

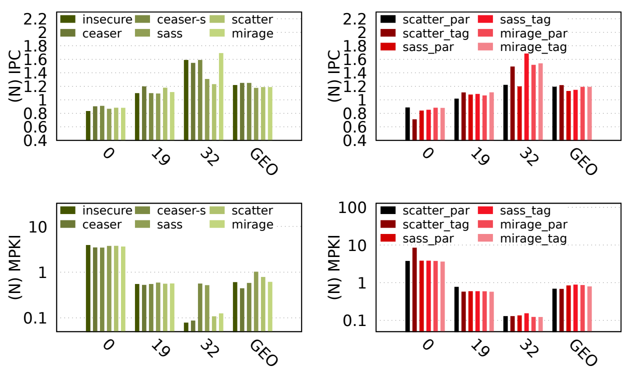

Figure 2: Three workload mixes with different performance levels ranging from low, medium to high, and their correlated results to the MPKI metric (see Appendix §A)

A PERFORMANCE OF SPECRATE 2017 (WITH NO DATA SHARING)

We generate a set of workloads that exhaustively combine different kinds of benchmarks with different LLC characteristics. In our first set of simulations, we use an 8-core system, which has 8 simulated processors running simultaneously. We generate 35 different combinations of benchmarks, where 8 benchmarks are running simultaneously. In order to maximize the number of security domains, we assign different security domains to different benchmarks. Hence, there are 8 security domains running concurrently. Two key performance metrics are the overall *instructions-per-cycle* (IPC) and the overall *misses-per-kilo-instruction* (MPKI). The first metric is very important because it represents the throughput of the system. The second metric is also important because it represents as a number whether the cache is performing well or not.

Performance Results: We show the performance results for the 8-core SPEC benchmarks in Figure 1. The figure is split into two parts, with the first six configurations in the left half and the next six configurations in the right half of the figure. The performance is sorted according to increased normalized IPC. We see that the performance of the randomized caches is generally very close to that of the *insecure* cache. The *sass* configuration has the worst performance of the lot, which is as expected due to the logical partitioning restricting the amount of space used in the LLC for each security domain. In the right half of the figure, we see performance results for the other 6 configurations, with either the parallel cache accesses or the tag-based filter added. In these configurations, the main change is that latency is increased, which causes the performance to be slightly worse.

In Figure 2, we observe the IPC of three workloads with different performances, and observe that the MPKI trends are roughly in the opposite direction of performance. Hence, we have evidence that the cache performance is has significant impact on the overall performance of the workloads.