# Machine Learning technique to improve performance of cache bypassing

Warisa Sritriratanarak and Prabhas Chongstitvatana
Department of computer engineering, Chulalongkorn University
prabhas.c@chula.ac.th

## Abstract

As the processor-memory performance gap increased, the multilevel cache is introduced. For example, the Intel Pentium processor has level 1 on-chip cache and level 2 off-chip cache implemented; the L2 cache moved up to be on-chip in the Intel Pentium Pro. With the advent of multicore processors, several private and shared levels of cache are used, i.e., the Intel Core Xeon X5550 has four cores and three layers of cache: each core has private L1 and L2 cache; the Last-level cache (LLC) is shared among four cores. Sharing the LLC can reduce the number of duplicate copies when many cores are running identical applications.

This work concerns with non-inclusion cache which a datum can have multiple copies like an inclusive cache but when a datum is discarded from the lower level cache, the copies in the upper levels are not required to be discarded too. When the datum moves up from lower level to higher level, the exclusive cache will force deletion of the copy in lower level while the non-inclusive cache will allow leaving the copy in lower level, avoid wasting any blocks.

Many workloads today are multimedia applications which load millions of blocks of data and use them only once. Some data are placed in cache and never reused again until they are evicted from cache. This is especially true in the LLC, since exploitation of temporal locality in high level cache means an inversion of temporal locality on LLC; in other words, data that is accessed frequently will always hit on level 1 caches, thus remain unused on levels 2 and 3. Study shows that numerous data blocks are allocated to the last-level cache and never reused or accessed again. Those blocks that are placed in the LLC and never accessed should never be allocated on cache to waste precious cache space. The method of not allocating some data to cache is called cache bypassing.

Most bypass techniques have relied on ad hoc methods such as counters and tables which cannot tackle the complexity of multicore workloads. In this work, we propose an alternative method to predict cache bypassing using Support Vector Machine (SVM) models. Based on access traces obtained from representative benchmarks running on the Multi2Sim simulator, supervised SVM training was performed in order to obtain a bypass prediction model suitable for LLC in multi-core processors. The SVM outputs bypassing classifiers which are integrated on the simulator to quantify LLC performance improvements. Results show that, with appropriate parameters and kernel functions, SVM is capable of generating bypassing models which improve LLC performance on multicore processors, achieving an average 5.34% hit rate improvement across SPLASH2 benchmark combinations.

keywords: machine learning, last level cache, cache bypassing