Randomized Last-Level Caches Are Still Vulnerable to Cache Side-Channel Attacks! But We Can Fix It

Hyperthreads (**6**)
Sumon Nath
(21Q050007)

I. Summary

The authors show that state-of-the-art cache randomization techniques like CEASAR-S and ScatterCache make flawed assumptions which renders them vulnerable to conflict-based side channel attacks. They propose a few techniques as well as use some existing techniques to fix these vulnerabilities. The experimental results show that these new mitigation techniques does not have a large performance hit compared to the existing techniques. They also argue that randomized set-associative caches without any partitions are a reasonable choice for commercial systems as compared to their skewed counterparts by showing that they can be strengthened enough to mitigate the aforementioned attacks.

II. Details

A. Problem with existing techniques

- Small eviction sets with partially congruent addresses can be found (by algorithms like Conflict Testing) in a much lesser time window compared to the remap period of randomized caches.
- These small eviction sets have an eviction rate(probability of evicting the target address) of as low as 30%. But these evictions sets can lead to a high overall eviction rate (close to 90%) by using a prime-probe-flush mechanism repeatedly. This mechanism is proposed by the authors.
- Existing mitigation techniques ignore the facts that (1) eviction sets with partially congruent addresses can be used to launch effective attacks and (2) eviction sets with low eviction rate can be repeatedly flushed and re-accessed to get an overall high eviction rate.

B. Fixes proposed

- They propose to define the remap period of randomized caches in terms of LLC evictions rather than using LLC accesses. The reasons being (1) LLC accesses are filtered by higher-level cache accesses and (2) LLC evictions are much higher during an attack as compared to normal execution. Thus LLC evictions are a much better quantification for setting the remap period.
- They propose a mathematical model to estimate the number of evictions required to find a small eviction set(explained in section II.A) to set the remap period appropriately to avoid creation of eviction sets by attacker.
- They use ZCache a multi-step relocation technique to remap blocks in a randomized cache to reduce the number of LLC evictions in the remap process and hence reduces performance degradation.

C. No Skewed caches

- The problems with skewed caches are, (1) they complicate hardware design of LLC and (2) incurs high performance degradation in case of high skewing. So the authors propose to strengthen normal caches instead.
- They propose an attack detection mechanism for non-skewed caches which can stop algorithms like Prime, prune & test (PPT) and Group elimination(GE) to find eviction sets by remapping the randomized cache, whenever an attack is detected.
- They use a mathematical model based on Zscore for each cache set to detect an attack. Whenever the Zscore crosses some threshold (which is set carefully), an attack is detected and the cache is remapped.

III. Positives

- Assumptions regarding attackers are highly pessimistic in nature and thus the attack model is very strong. Mitigation against such an attack model will be hard to break.
- They validate the experiments for probability of finding eviction sets by attackers and compares them with a mathematical model. They find very close match between the experimental and mathematical results.
- The proposed idea to set remap period in terms of LLC evictions significantly reduces LLC MPKI by 69% and the number of remaps by 64% as compared to when LLC accesses are used to set remap period. This reduces overall performance degradation for randomized caches.
- They use the Spike simulator which increases the number of simulated instructions by 100 to 400 times compared to other simulators like Gem5 simulator. This results in better coverage of the benchmarks.
- They propose a hardware solution with minimal hardware overhead to implement the multi-step relocation technique to remap cache lines and to detect attacks(explained in the section II.C).

IV. Negatives

- The authors did not gave a proper explanation of how the small eviction sets with partially congruent addresses can yield a high eviction rate by repeatedly flushing and re-accessing the set (explained in Section II.A).

- For the above point they find the vulnerability but does not propose a mitigation for it.
- First they propose ideas to fix skewed caches. But in the later sections of original paper they say skewed caches should not be used as they incur high performance degradation, which makes the proposed fixes for skewed caches incoherent with the later argument of not using them at all.
- In section IV.B of the original paper, authors show that CT algorithm is faster than PPT and GE, but in section VI.B they find PPT and GE is faster than CT, but do not give a proper reasoning as to why this happens.
- In section VI.B of the original paper, the authors say that the remap period for non-skewed caches is short enough and instead of reducing them further they propose an attack detection mechanism to trigger remaps. But they do not give a proper reasoning of why not to reduce the remap period further.
- They find out appropriate remap period for different kinds of caches using a mathematical model to avoid attacks. But in section VI.B of the original paper, they say that the remap period is not long enough to mitigate attacks, which is contradictory to their previous findings.
- A minor issue, the strong attack model used may be a weakness in the sense that previously proposed mitigation techniques may not be secure enough against such an attack model, even though such a strong attack model is highly improbable in practical scenarios. For example, they assume there is no noise in the side channel, which is a bit superfluous.
- They do not include geomean/mean in the plots which makes them a bit difficult to comprehend.

V. Possible extensions

- The technique proposed to detect attacks only target few eviction set finding algorithms like CT, PPT and GE. There is scope of evading these mitigation techniques by introducing new algorithms to find eviction sets. So extending the mitigation techniques to encompass more attacks can be a possible extension.
- A possible Denial of service(DOS) attack can be made, where the attacker searches for eviction sets frequently enough to fool the attack detection technique proposed by the authors to frequently remap the randomized cache incurring huge performance penalty.