Assignment II – AnC

Team quench-quokka
Mihai-George Licu
Changling Wang

AnC - Overview

- Address Space Layout Randomization -> puts code at random virtual memory addresses
- When the MMU translates virt -> phys address it incurs a page table walk
- Page table entries are cached in the CPU LLC
  - Cache line has 64 bytes
  - Each page has 64 distinct non-overlapping cache lines -> Accessing cache line #i of many pages evicts all sets that could map cache line #i of any page
- Evicting correct cache sets and timing -> find out what cache sets hold page table entries
- Sliding by memory offsets based on page map layer page size identifies PTE

AnC - Deliverables

- ./test__evict_cache - time access before and after eviction for all 64 cache lines of a memory page
- ./test__evict_tlb - time access of an address before and after evicting it **only** from the TLB
- ./anc - takes a virtual address and leaks it's derandomized virtual address
- heatmaps for PML1 and PML234

Step 1 - Evict TLB

- Evict everything (i.e. both cache & tlb), using the solution for Task2
- Read another entry (far from the target entry) to bring cache back.

Step 2 - Evict Cache Sets

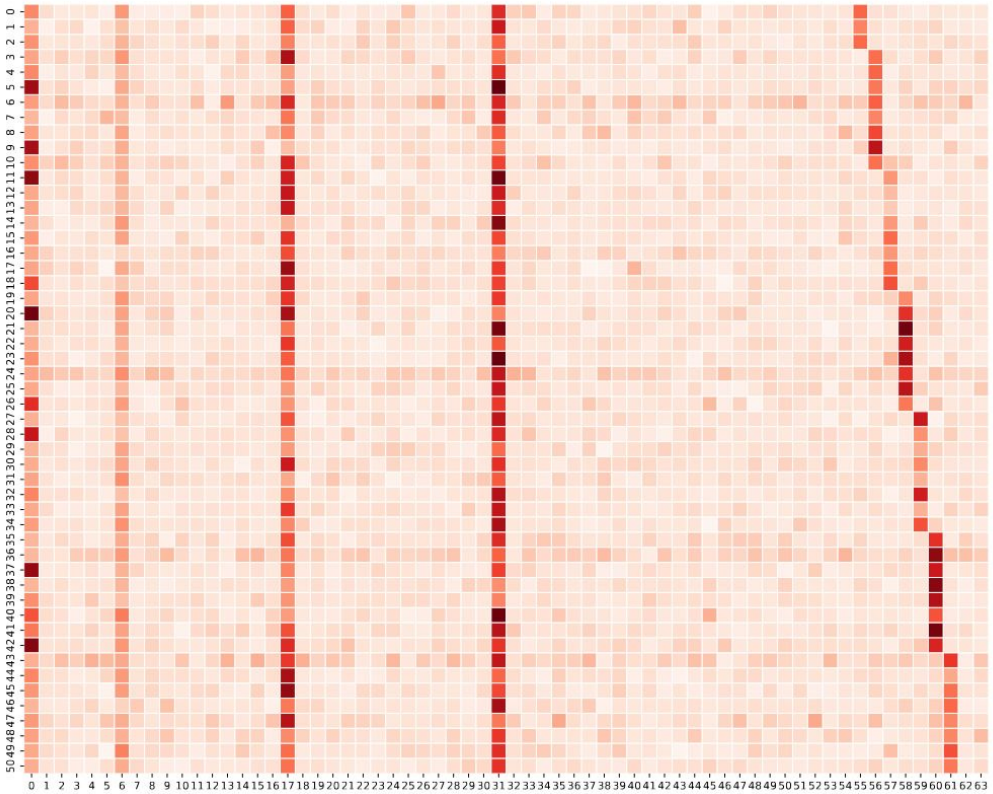
- Eviction set size varies for different levels of PM
- 24MB for PML1 (4x STLB size of victim CPU)
- 128MB for PML2
- Measure latency over 10 iterations, and subtract maximum before averaging. (improves stability)
- Do the measurement of cache lines out of order to mitigate noise from hardware prefetching

```
for (int set = 0; set < 64; ++set) {
  int idx = ((11 + 2 * i) + (13 + 2 * i) * set) % 64;
```

Step 3 - PML1 and heatmap

Timing all cache lines of a memory page

Increment the offset by 8 * pml size each time to see the staircase



Step 4 - identifying staircase

- “Convolutional Edge detection” inspired by image edge detection

```
for (int i = 0; i < 64; ++i) {
  times[i] = (times[i] - maxs[i]) / (round - 1);
  convolution[(64+i-offset)%64] += times[i];
  threshold += times[i];
}
```

Step 5 - Sliding

- Binary search works if signal is stable
- O(N) -> O(logN), though N is only 8