SCA 04

①

Core

ITLB DTLB

↓ ↓

STLB

↓

PDE Page Directory Entry

↓

PDPTE Pointer Table Entry

↓

PML4E

↓

Page Table structures in caches

↓

"—" in DRAM

②

- Constant-Time Techniques:
  Ensure Code behaviour isn't data dependent.
  The sequence of cache accesses/branches isn't dependand
  Performance Cost but practicable
- Hardware Support:
  Avoid vulnerable table lookups by providing constant time hardware operation,
  Language based: semantics with no variant execution length
- Injecting noise:
  "Fuzzy Time": Inject noise into ~~alle~~ all events, visible to a process,
  "Random Permutation cache": randomized indexes
  ⇒ Randomly evicts
  ~~Little Cost~~ Enough noise would slow performance.

- Enforcing Determinism:
  Eliminat visible timing variation
  - no real time, only virtual clocks, independent of actions by vulnerable components.
    => 30% Overhead
  - Black-Box: control time of externally visible events
    => only for remote attacks

- Partitioning time:
  concurrent / consecutive access attacks, counter by time sliced exclusive access or managing time-slice transition.
  - Flush on CPU switches: for L1 ok, for lower to slow
  -

- Partitioning Hardware:
  Partition hardware resources between competing threads / cores
  - Disable Hyperthreading, Page sharing
  - L1 between threads
  - Cache colouring: Prevents different processes from accessing the same cache.
    → no large pages
    → performance improvement

- Auditing: Detect by monitoring
  ⇒ doesn't defend

③