Im in ur Cache Keepin ur Bitez

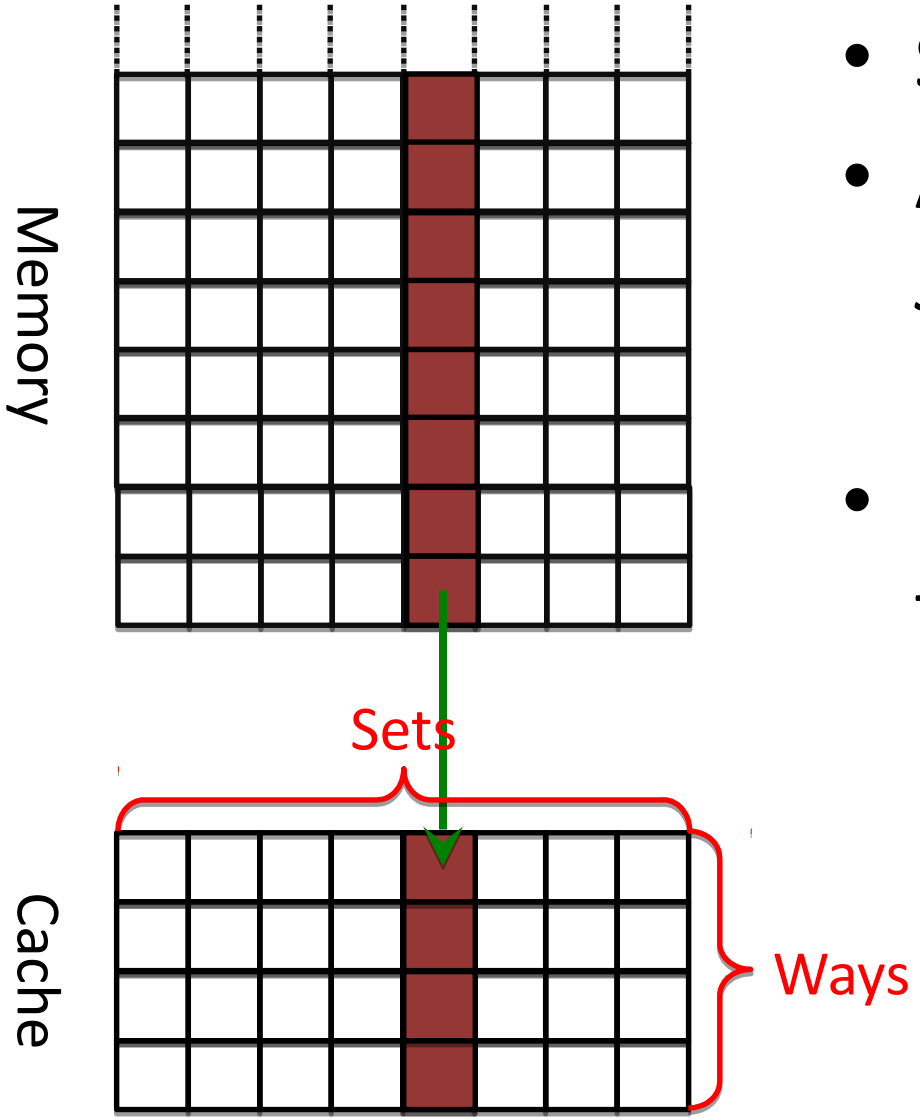
CATalyst: Defeating Last-Level Cache Side Channel Attacks in Cloud Computing

Fangfei Liu
Qian Ge
Yuval Yarom
Gernot Heiser
Ruby B. Lee

LLC (Last-Level Cache) Attacks

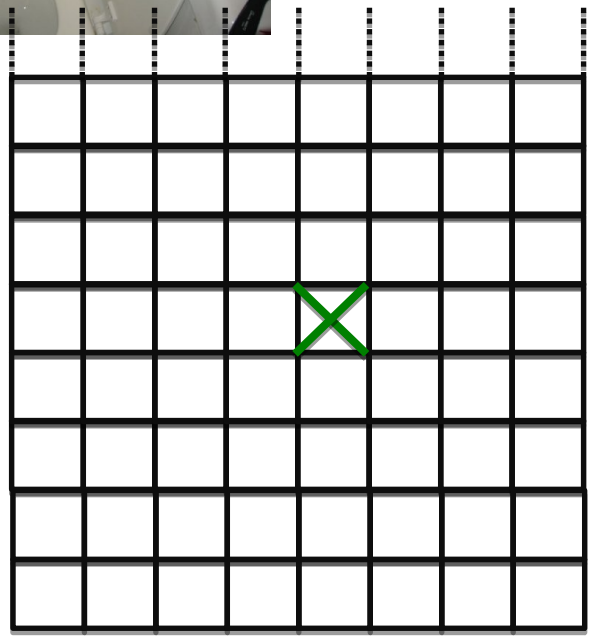
- Flush+Reload
  - GnuPG RSA [YF'14]
  - OpenSSL AES [IIES'14]
  - OpenSSL ECDSA [BPSY'14, PSY'15, ABF+'15]
  - Lucky 13 [IIES'15]
- LLC Prime+Probe
  - GnuPG RSA [LYG+'15]
  - OpenSSL AES [IES'15]
- **We want some countermeasures!**

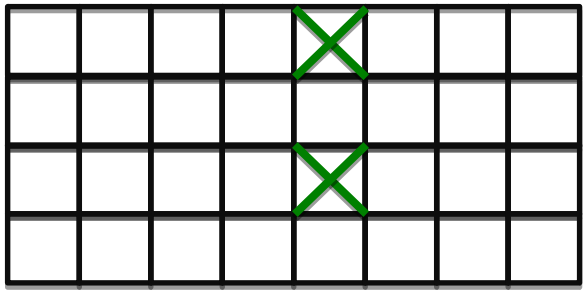
Cache Structure



- Stores fixed-size *lines*
- Arranged as multiple *sets*, each consisting of multiple *ways*.
- Each memory line maps to a single cache set
  - Can be cached in any of the ways in the set

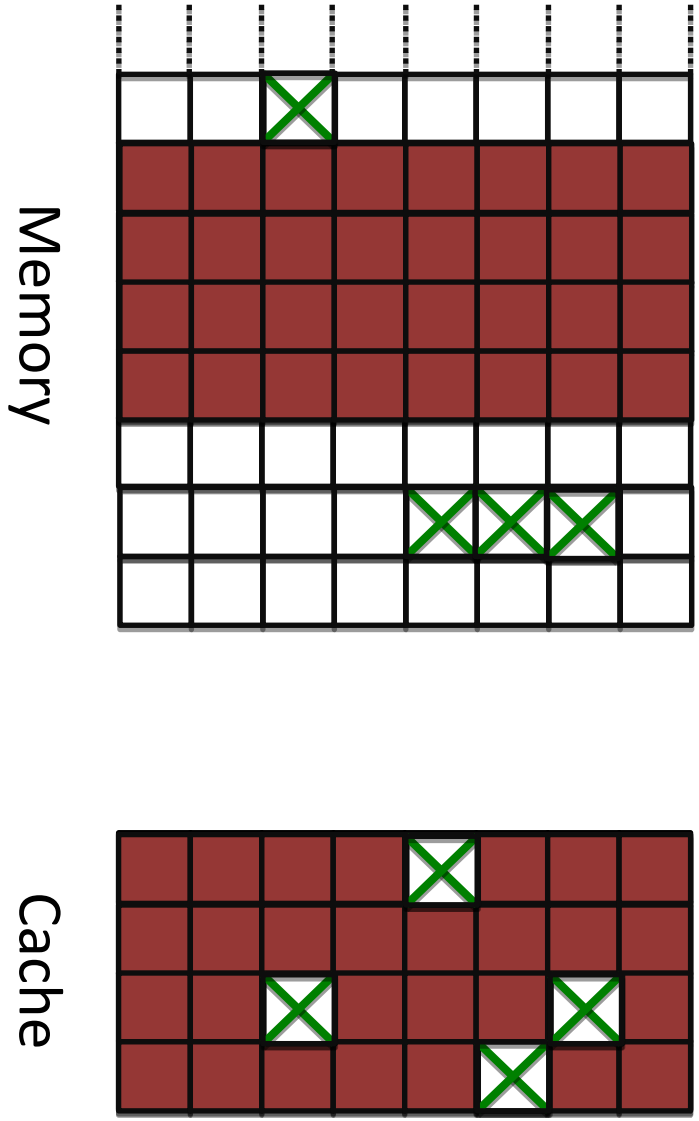
The Flush+Reload attack [YF'14]





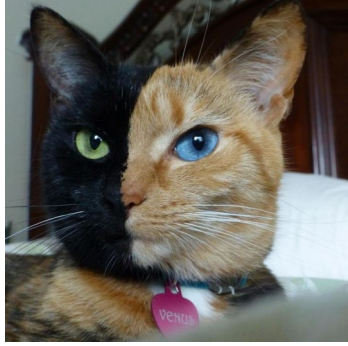
- Evict a **shared** cache line from the cache
- Victim executes, possibly accessing the shared line and caching it
- Measure time to access the shared line
  - Access to cached lines is faster than to evicted lines

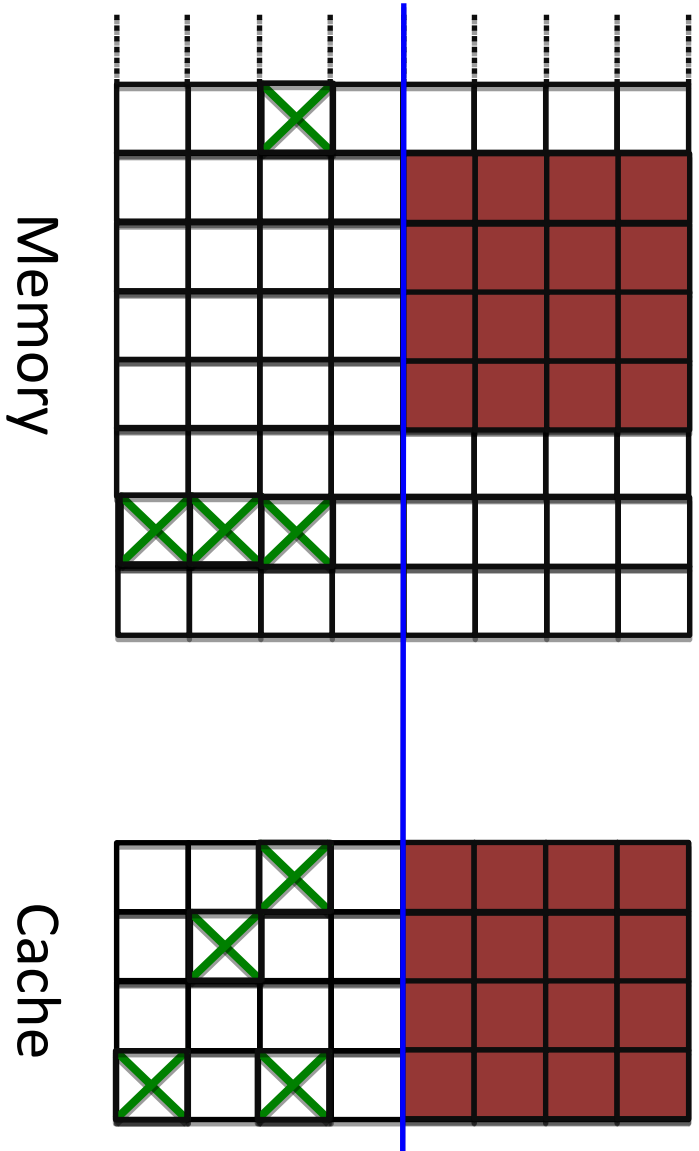
The Prime+Probe attack [Per'05,OST'05]



- Choose a cache-sized memory buffer
- Access all the lines in the buffer, filling the cache
- Victim executes, evicting some of the buffer lines from the cache
- Measure the time to access the buffer
  - Accesses to cached lines is faster than to evicted lines

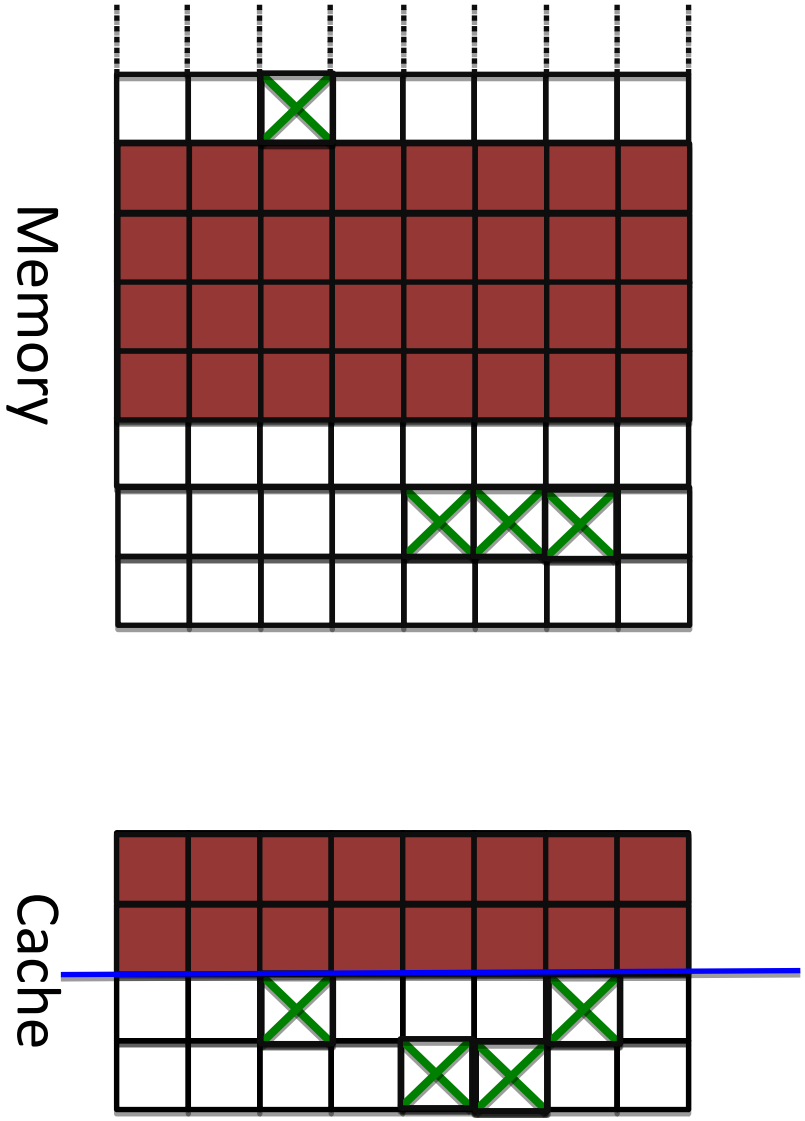
Solution 1: Page Colouring [Page'03]





- Partition cache sets.
- Relies on virtual memory to map process's pages to part of the physical space
- Limited number of colours (32 on Intel)
- No huge pages
- Rigid memory allocation

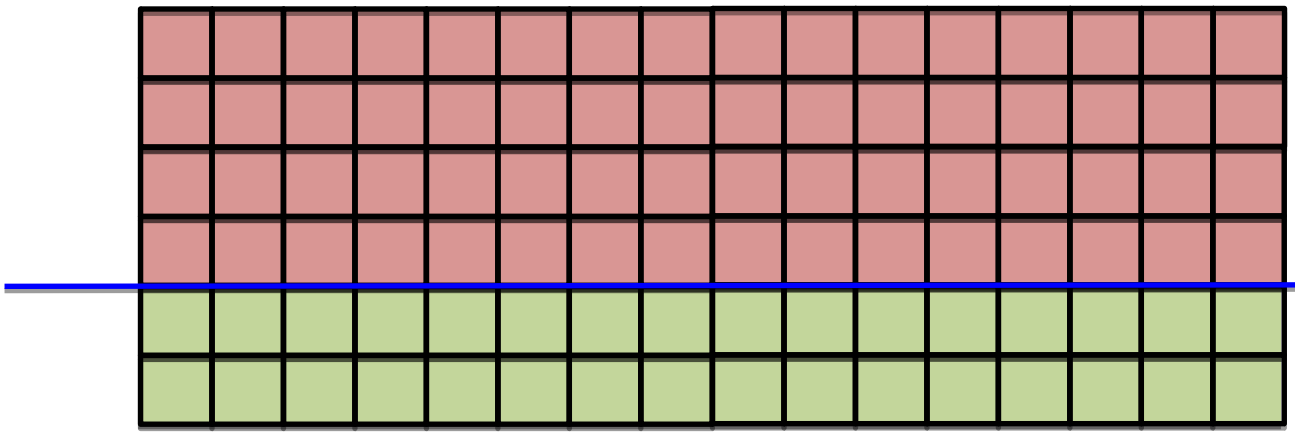
Solution 2: Split by ways



- NoMo caches [DJL+'12]
- Intel Cache Allocation Technology (CAT)
  - 4 Classes of Service (COS)
  - Restricts replacement

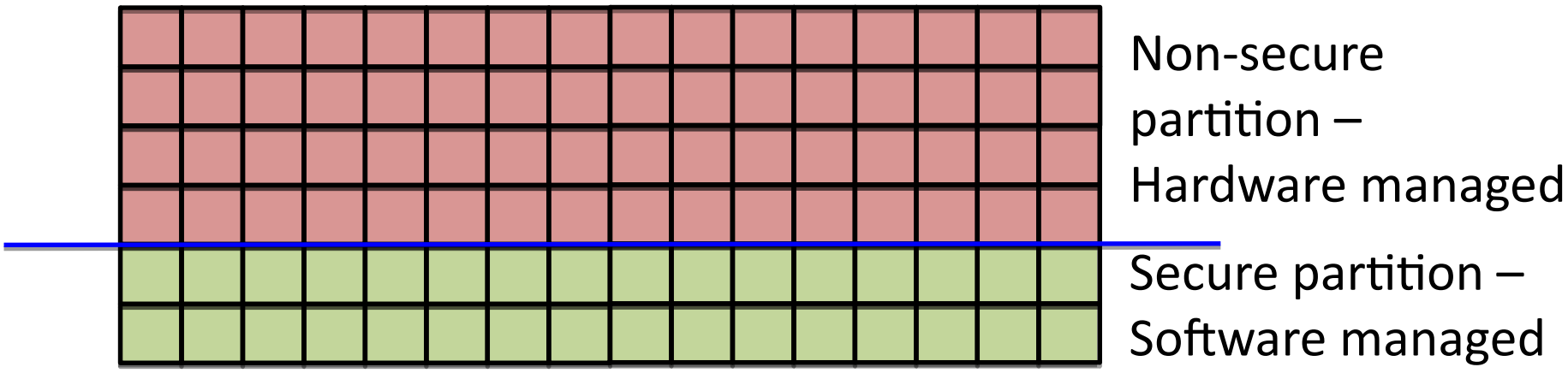
CATalyst

- Solution for virtualised environments
- Secure pages
  - Not shared (no Flush+Reload)
  - Pinned in the cache (no Prime+Probe)



LLC pseudo-locking

- At boot: preload secure pages into LLC secure partition
- At runtime: use CAT to limit replacement to non-secure partition
- Result: secure pages pinned in the secure partition

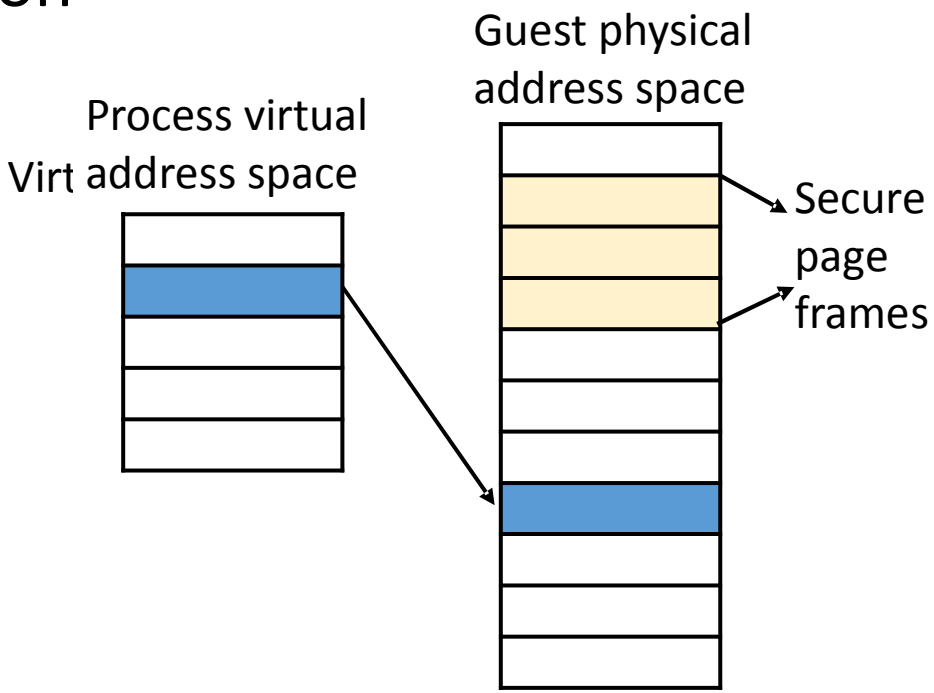


Preloading secure pages

- Non-secure partition (COS 0) mask `0x3ffff`
- Secure partition (COS 3) mask `0xc0000`
- Algorithm:
  1. Disable interrupts
  2. Access all of the cache lines of the preloader code
  3. Access one word of the preloaded page
  4. Set current processor to COS3
  5. CLFLUSH every cache line of the preloaded page
  6. Access every cache line of the preloaded page
  7. Set current processor to COS0
  8. Enable interrupts

Using secure pages

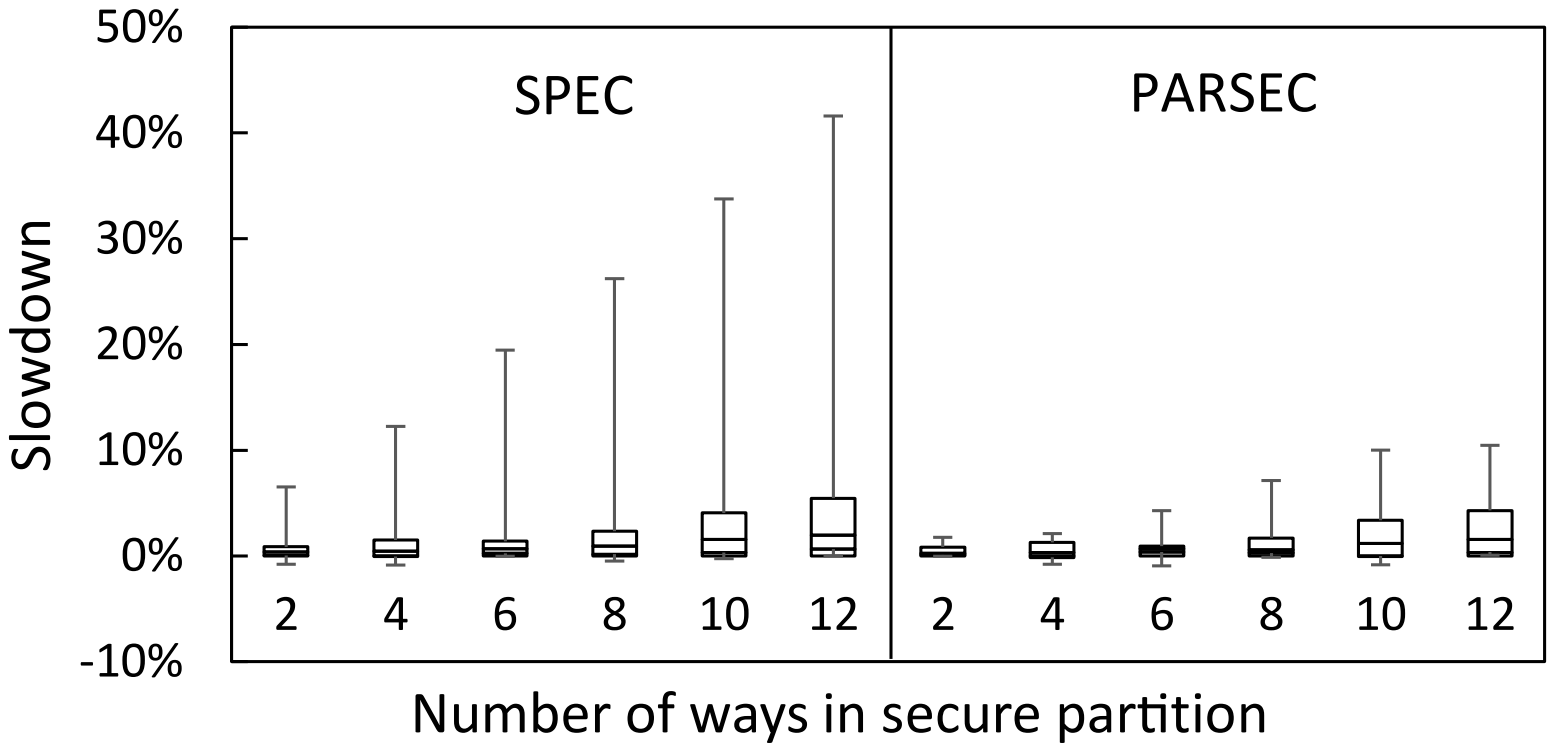
- Secure pages allocated to VMs at launch
- System calls map and unmap secure pages to processes
- Secure pages purged and reloaded on VM termination



Possible Issues

- Cache evicting instructions
  - Whole cache invalidation causes VM exits
  - Flushing single lines – requires access to the evicted lines
- Direct Data I/O
  - Cache access controlled via an MSR
- Cache coherence in multi-socket systems
  - Use CPU affinity to avoid cross-socket access to secure pages

Performance – legacy apps



Performance - encryption

Buffer size	5KiB	50KiB	500KiB	5MiB
Baseline(ms)	81.04	79.91	79.83	82.87
CATalyst (ms)	95.26	81.96	79.96	82.83
Slowdown	17.54%	2.56%	0.16%	-0.05%

AES – Encrypt 5 MiB

Version	1.4.13	1.4.18
Baseline (ms)	15.36	12.77
CATalyst (ms)	15.40	12.83
Slowdown	0.25%	0.44%

GnuPG – ElGamal Decryption Times

Summary

- CATalyst is a hybrid hardware/software managed cache
  - Hardware managed non-secure partition
  - Software managed secure partition
- Secure pages pseudo-locked in the secure partition
  - No information leaks through access patterns to secure pages
- Low performance overhead