Taming the Incoherent Cache Issue in Confidential VMs

Jacky Li <jackyli@google.com>
Mingwei Zhang <mizhang@google.com>
● Problem Statement
  ■ Incoherent cache lines
  ■ Performance degradation
● Solution: Selective Cache Flushing
  ○ MMU Notifier
    ■ Introduction
    ■ filtering the reason
• **C-bit**: mark whether a memory page is encrypted.

Incoherent cache lines

<table>
<thead>
<tr>
<th>C-bit</th>
<th>Cache Tag</th>
<th>Set Index</th>
<th>Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>...</td>
<td>010</td>
<td>00010</td>
</tr>
<tr>
<td>1 0 0</td>
<td>...</td>
<td>010</td>
<td>00010</td>
</tr>
</tbody>
</table>

• Memory Management (kernel) **recognizes** the C-bit
• Cache (hardware) **doesn't know** about C-bit
CVM releases the page -> non-CVM gets the same page

- 2 conflicting cache lines => Data Corruption
- Solution (2017): flush cache[1]
Performance degradation

- SME_COHERENT (2020): CPU cache recognized the C-bit so no need to flush. [2]
  - Vulnerability CVE-2022-0171.
    - ✅ CPU => CPU
    - ❌ CPU => DMA devices
  - Solution: cache flush in mmu notifier when the page leaves CVM [3]
    - Perf impact [4]
Solution:

Selective Cache Flushing

- Cache Flush **Only When** VM deallocate memory?
  - We are trying to...
  - KVM MMU does not manage memory...
  - We have to do it at MMU_NOTIFIER

- Cache Flush in **SMALLER** Granularity?
  - We wish...
  - Non-trivial changes on KVM.
  - Non-trivial changes on MM.

Note: this problem will only affect SEV/SEV-ES
**MMU Notifiers: reasons**

MMU notifier invalidation does contain a reason parameter which is unused currently.

KVM MMU represents **guest VM**

Host MMU represents the **process**

**mmu_notifier_event:** This is why I am asking you to ZAP your mapping!

**mmu_notifier_invalidate_range_start()**

**mmu_notifier_invalidate_range_end()**

**Host OS**

**MMU notifier invalidation does contain a reason parameter which is unused currently.**
MMU Notifier: filtering the reason

- MMU_NOTIFY_UNMAP
- MMU_NOTIFY_CLEAR
- MMU_NOTIFY_PROTECTION_VMA
- MMU_NOTIFY_PROTECTION_PAGE
- MMU_NOTIFY_SOFT_DIRTY
- MMU_NOTIFY_RELEASE
- MMU_NOTIFY_MIGRATE
- MMU_NOTIFY_EXCLUSIVE

Questions:
- munmap?
- madvise? migrate?
- process died?
- NUMA balancing?
MMU Notifier: filtering the reason

- Flush cache selectively on mmu_notifier is the most cost effective approach with minimum changes to KVM

- We have had discussions with AMD about addressing this issue in future HW
Thank You!

Q&A

Jacky Li <jackyli@google.com>
Mingwei Zhang <mizhang@google.com>
Appendix:

[1] 89c505809052 (“KVM: SVM: Add support for KVM_SEV_LAUNCH_UPDATE_DATA command”)
[2] e1ebbb2b49048 (“KVM: SVM: Don’t flush cache if hardware enforces cache coherency across encryption domains”)
[4] https://lore.kernel.org/kvm/YzJFvWPb1syXcVQm@google.com/T/#mb79712b3d141cabb166b504984f6058b01e30c63
MMU Notifiers: reasons

KVM MMU represents guest VM

Host MMU represents the process

MMU notifier is the memory reclaim interface between KVM and host MM

Cache flush is done here!

mmu_notifier_invalidate_range_start()

mmu_notifier_invalidate_range_end()
VM_PAGEFLUSH: limited functionality

- MSR_AMD64_VM_PAGE_FLUSH (0xc001011e)
  - CPUID level 0x8000001f (EAX), bit 2
  - X86_FEATURE_VM_PAGE_FLUSH
  - Available on AMD EPYC v1 and later

- VM_PAGEFLUSH MSR does not work on user addresses
  - Even if we disable SMAP (EFLAGS.AC)
  - AMD APM updated on this at the end of 2022