Linux
Plumbers
Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Identifying and Eliminating Contention from Booting Concurrent SNP VMs

Jacky Li <jackli@google.com>
Marc Orr <marcorr@google.com>
Alper Gun <alpergun@google.com>

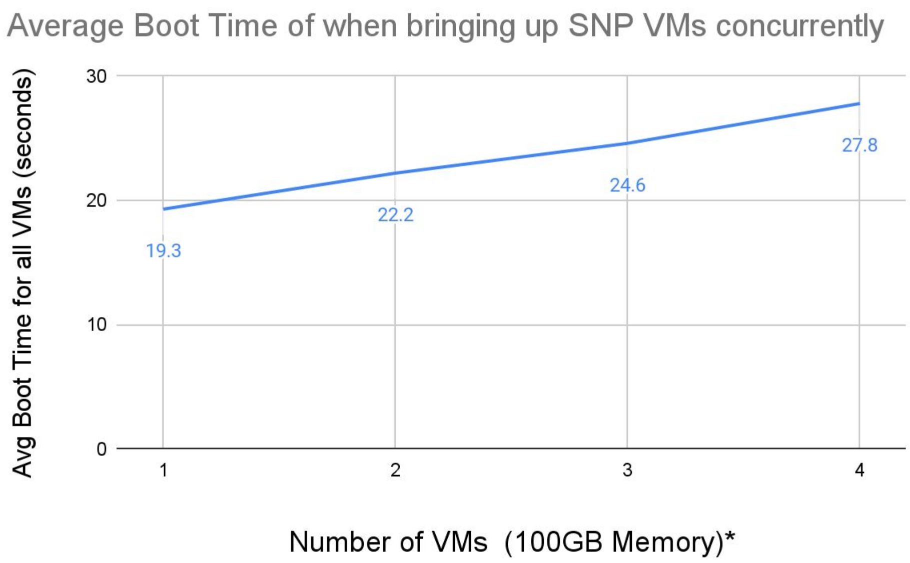
- Problem Statement
    - Booting SEV-SNP VMs concurrently
    - Identifying the Contention

- Solutions
    - Removing Lock Contention
    - Rate Limiting Page State Change (PSC) requests from the guest.

Problem Statement

Booting SEV-SNP VMs concurrently

- SEV-SNP[1] (Secure Encrypted Virtualization - Secure Nested Paging)
    - memory encryption tech by AMD
- Boot Time investigation
- What happens?
    - Initiating RMP entries in a loop.

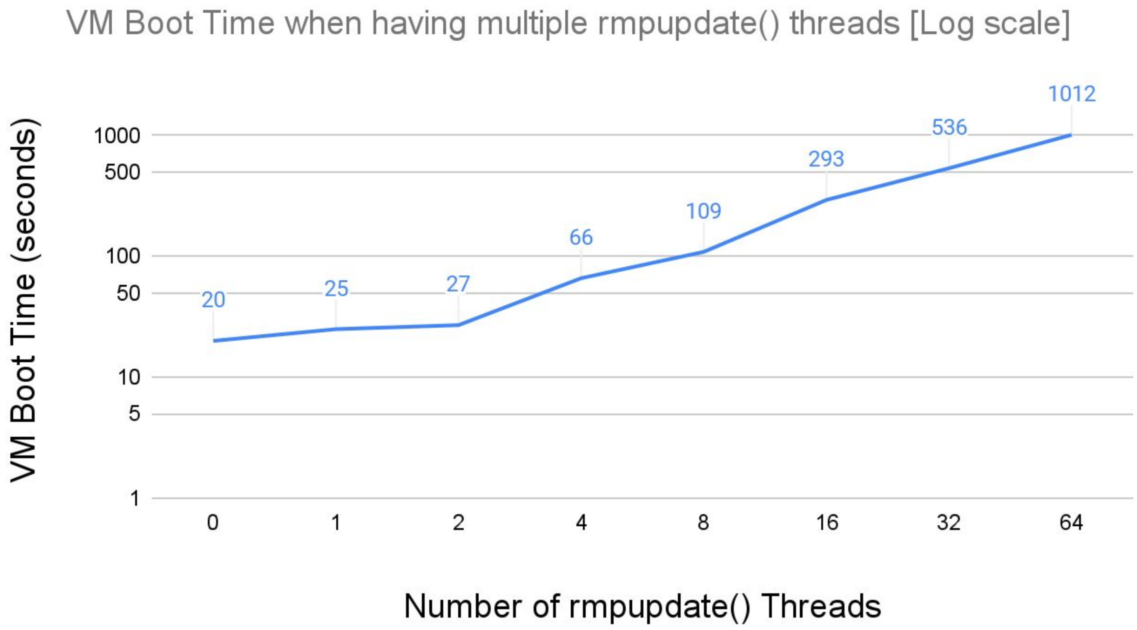


* Kernel: 5.15 LTS + SNP v5 patches, CPU: Zen 3 (milan)

Problem Statement

Booting SEV-SNP VMs concurrently

- RMP[2] (Reverse Map Table) - track owner for every page in memory.
- Further Investigation on rmpupdate() function
- What happens?
    - rmpupdate() contention!



Problem Statement

Identifying the Contention

```
rmpupdate()
    if (page_getting_assigned_to_guest): set_direct_map_invalid();
    do_asm_rmpupdate;
    if (page_returned_to_host):  set_direct_map_default();
```

- Both set_direct_map_xxx() call into __change_page_attr() which is protected behind a global spin lock (cpa_lock) for every single request to change attribute.
  - Doesn't smells right because we are changing different addresses/pages.

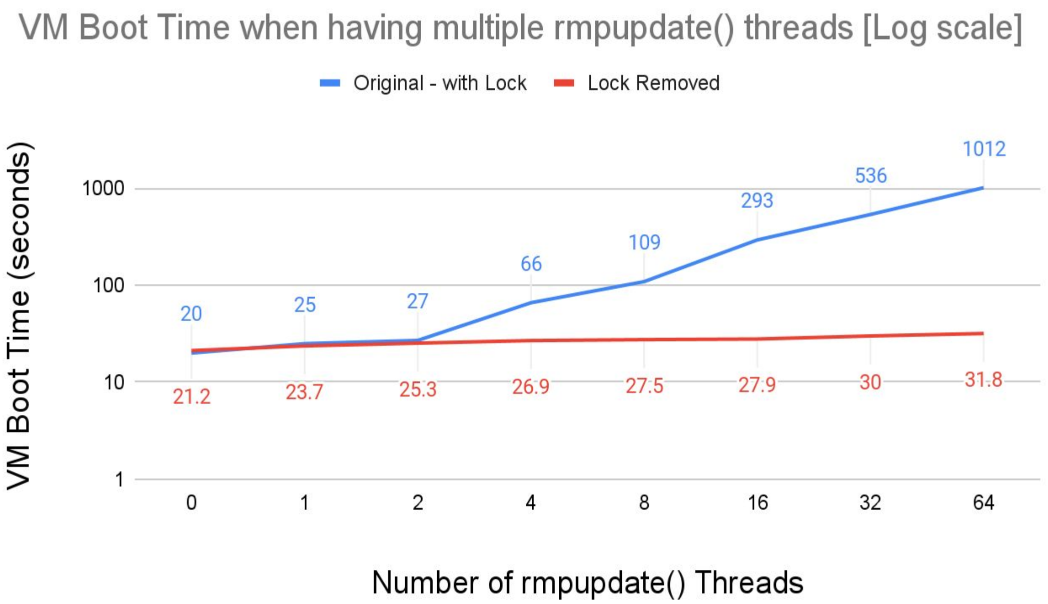
Solutions

Removing Lock Contention

- Before digging more into the necessity of the lock, let's remove it first.

- No crash, no misbehaved function, yet better performance

- Now, can we really remove it?



Solutions

Removing Lock Contention

- We think we can.

- The cpa_lock is introduced in 2008[3] for solving a race condition:

CPU-0 is splitting the large page && changing attributes

CPU-1 is changing attributes

  - *“When it happens, Intel CPU has undefined behavior and can use neither of the translation in TLB.*

Solutions

Removing Lock Contention

- This race condition itself has been protected by another global spin lock (pgd_lock) TODAY
  - Thanks to a patch in 2018 that moves tlb_flush

Code in 2008

```
pgd_lock;
if change_attribute_large_page:
    __set_pmd_pte();        <- CPU-1 program counter
pgd_unlock;
if should_split:
    pgd_lock
        split_large_page();
    pgd_unlock              <- CPU-0 program counter
    tlb_flush
    change_attribute_4K_page
```

Code in 2018

```
pgd_lock;
if change_attribute_large_page:
    __set_pmd_pte();        <- CPU-1 program counter
pgd_unlock;
if should_split:
    pgd_lock
        split_large_page();
        tlb_flush           <- CPU-0 program counter
    pgd_unlock
    change_attribute_4K_page
```

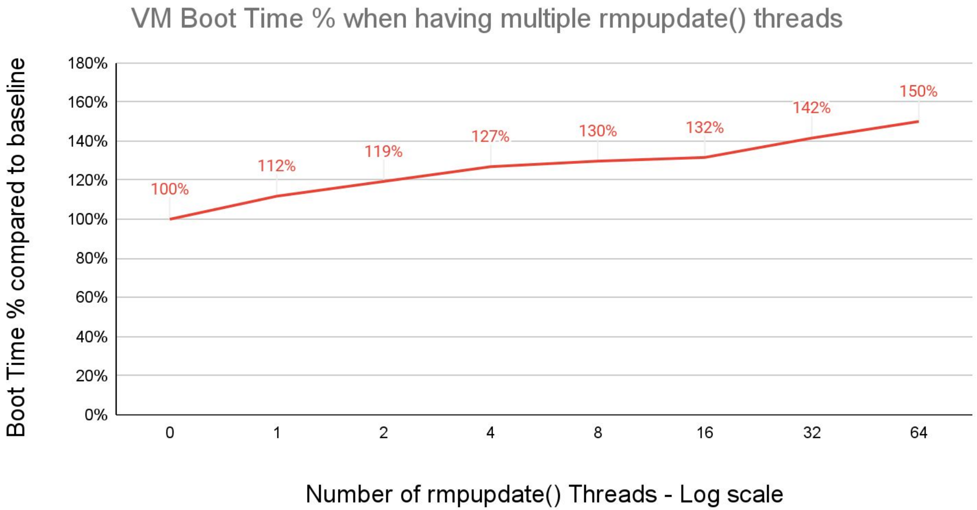
Solutions

Rate Limiting Page State Change requests from the guest

- Even after removing the cpa lock, boot time still degrades

- Further optimization against pgd_lock
  - per-PMD lock[5]

- Generic: hardware overcommitted
  - Rate/Quota limit enforcement
  - Lazy accept



Discussions

- We will be sending out the patch to remove the cpa_locks soon
  - It would be great to collect feedbacks in this talk
- Collecting thoughts on optimizing PGD lock to per-PMD lock.
- Collecting thoughts on handling hardware resource overcommitment systematically.

Linux Plumbers Conference 2022
>> Dublin, Ireland / September 12-14, 2022

Acknowledgement

We thank all the helps along the way that make this talk possible

- David Kaplan, for helping investigation and providing feedback.
- Brijesh Singh, for helping investigation and sharing testing patches.
- Frank van der Linden, for helping clarifying the race condition and suggesting removal of the lock
- David Rientjes, for encouraging us to do the work and connecting us to the right people

Appendix

- [1] SEV-SNP: SEV-SNP builds upon existing SEV and SEV-ES functionality while adding new hardware-based security protections. SEV-SNP adds strong memory integrity protection to help prevent malicious hypervisorbased attacks like data replay, memory re-mapping, and more in order to create an isolated execution environment. Also, SEV-SNP introduces several additional optional security enhancements designed to support additional VM use models, offer stronger protection around interrupt behavior, and offer increased protection against recently disclosed side channel attacks.
    - SEV-SNP whilte paper: https://www.amd.com/system/files/TechDocs/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf
- [2] RMP: many of the integrity guarantees of SEV-SNP are enforced through a new structure called the Reverse Map Table (RMP). The RMP is a single data structure shared across the system that contains one entry for every 4k page of DRAM that may be used by VMs. The goal of the RMP is simple: it tracks the owner for each page of memory. Pages of memory can be owned by the hypervisor, owned by a specific VM, or owned by the AMD-SP. Access to memory is controlled so only the owner of that page can write it. The RMP is used in conjunction with standard x86 page tables to enforce memory restrictions and page access rights.

Appendix

- [3] 2008 Patch introducing cpa_lock: https://lkml.org/lkml/2008/9/23/385
  - ***"The TLBs may contain both ordinary and large-page translations for a 4-KByte range of linear addresses. This may occur if software modifies the paging structures so that the page size used for the address range changes.*** *If the two translations differ with respect to page frame or attributes (e.g., permissions), processor behavior is undefined and may be implementation specific. The processor may use a page frame or attributes that correspond to neither translation; it may improperly set or fail to set the dirty bit in the appropriate paging-structure entry."*
  - *"We do this global tlb flush inside the cpa_lock, so that* ***we don't allow any other cpu, with stale tlb entries change the page attribute in parallel****, that also falls into the just split large page entry."*
- [4] 2018 Patch moving tlb_flush: https://lkml.org/lkml/2018/9/19/339
  - *"There is an atom errata, where we do a local TLB invalidate right before we return and then do a global TLB invalidate. Move the global invalidate up a little bit and avoid the local invalidate entirely. This does put the global invalidate under pgd_lock, but that shouldn't matter."*

Appendix

- [5] Page Table Level Terminology Comparison (4-level)

Linux	PGD	PUD	PMD	PTE
AMD	PML4E	PDPE	PDE	PTE
Intel	PML4E	PDPTE	PDE	PTE