ORACLE

The Life Cycle of the mm_sturct

Liam R. Howlett
Linux Kernel Developer
December 11th, 2025

The mm_struct Friction Points

Recent Issue Causes:

- No VMAs in the task
  - Race with munmap or SIGSEGV and exit
- Failed to fully initialize
  - OOM, other forking issues leave incomplete task
- Exit racing with OOM
  - The PTE teardown with split PT locks causes slow down in races

The mm_struct Workarounds

- Use a Lock
  - Slower, trend is to avoid or limit lock time
  - OOM uses mmap read lock, usage is on the rise
- Tear down VMAs on failure
  - Still holds the mmap lock
  - Creates a “no VMA” task
- MMF_ bits (unstable, skip)
  - Not used globally, may not remove races


ORACLE

The Issues

fork: avoid inappropriate uprobe access to invalid mm

- perf locking issue
- details: Needed to move uprobe lock outside of dup_mmap() into dup_mm() and hold the lock on failure.

https://lore.kernel.org/linux-mm/20241210172412.52995-1-lorenzo.stoakes@oracle.com/

Do not delay oom reaper when the victim is frozen

- Reverse iterate to speed up exit (robust futexes)
- details: If a process holding robust futexes gets frozen, robust futexes might be reaped before futex_cleanup() runs when an OOM occurs.

https://lore.kernel.org/linux-mm/20250825133855.30229-1-zhongjinji@honor.com/

The Issues

kernel: Be more careful about dup_mmap() failures and uprobe registering

- Modify registering uprobes to check MMF_OOM_SKIP
- details: Add MMF_OOM_SKIP and MMF_UNSTABLE to dup_mmap() failure to avoid races with register_for_each_vma() in uprobe (and in oom code iterator).

https://lore.kernel.org/linux-mm/20250127170221.1761366-1-Liam.Howlett@oracle.com/

mm: swap: check for stable address space before operating on the VMA

- swap is swapping a failed dup_mmap() task and hitting the XA_ZERO_ENTRY
- details: Check for MMF_UNSTABLE by check_stable_address_space() to avoid dereferencing invalid VMA

https://lore.kernel.org/all/20250924181138.1762750-1-charan.kalla@oss.qualcomm.com/

The Issues

Remove XA_ZERO from error recovery of dup_mmap()

- Tear down the vma tree in the error path of dup_mmap()
- Details: When a failure occurs, a special entry is inserted to mark the location. This special entry is an issue when found for merging, etc, through other trees or the rmap. Still leaves an mm without any vmas, but the mmap_exit() path complicates the recovery.

https://lore.kernel.org/linux-mm/20250909190945.1030905-3-Liam.Howlett@oracle.com/

sched/numa: Fix the potential null pointer dereference in task_numa_work()

- Scheduler searches for a vma in an mm_struct, which may be null.
- Details: Change the loop to only run if there is a vma

https://lore.kernel.org/all/20241025022208.125527-1-shawnwang@linux.alibaba.com/T/#u


ORACLE