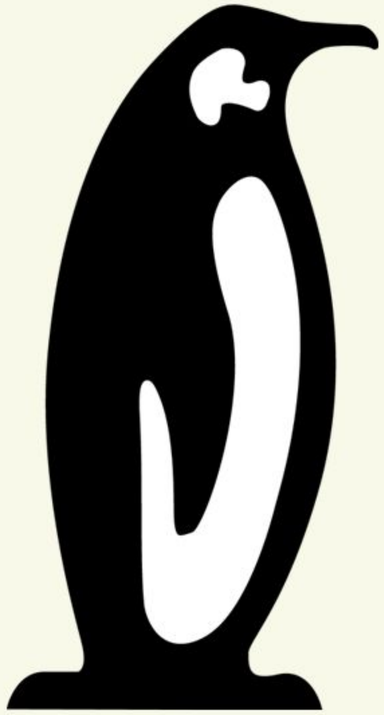


Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023

Linux
Plumbers
Conference | Richmond, VA | Nov. 13-15, 2023

Introducing PAGEMAP_SCAN IOCTL for Windows syscalls translation and CRIU

Muhammad Usama Anjum
Collabora

Andrei Vagin
CRIU

GetWriteWatch() and ResetWriteWatch() Windows APIs

- Get and/or Reset the write tracking state
- Operation on any size of memory area
- Returns after finding X dirty pages from total of N pages (X < N)
- API is relatively widely used on Windows
  - Implementing copy-on-write mechanisms
  - Security, intrusion and debugger detection
  - Garbage collectors

Translation by mprotect() + SIGSEGV or Userfaultfd() in Userspace

- Traveling of signals or messages between kernel and userspace
- Too much slow to be useful in high performance demanding applications like games
- Doesn't fulfill all the use cases
  - Write protected memory isn't desired by some software layers like Vulkan drivers

Soft-Dirty PTE flag and Soft-Dirty VMA flag

- Soft-dirty flag is a combination of PTE and VMA flag
  - If any of these two flags are set, the page is soft-dirty.
- Possible operations
  - Read soft-dirty flags of desired pages from pagemap file
  - Clear soft-dirty flags from *all* the PTEs of the process

Kernel's Internal Activity Affecting Soft-Dirty Flag

- VMAs having the different VMA soft-dirty flags are merged
  - This makes non-softdirty VMA soft-dirty
- If they aren't merged, there is possibility that the maximum vma limit is reached (/proc/sys/vm/max_map_count).

Shortcomings of Soft-Dirty Flag

- It is not accurate
- Atomic get and clear operation isn't possible
- Soft-Dirty flag on a part of memory region cannot be cleared

CRIU uses Soft-Dirty Flag

- CRIU freezes processes to pre-dump their memory
- CRIU doesn't have the accurate information about pages to the moment of dumping them
- CRIU struggles to handle huge sparse mappings

Add Prototype IOCTL based on Soft-Dirty Flag

- Implement atomic GET+CLEAR operation
- Implement Clear operation on a region of memory
- Optionally ignore Soft-Dirty flag on VMA
- The prototype got broken by upstream change:
  - The mprotect() stopped setting the Soft-dirty PTE flag when VMA Soft-dirty flag was set
  - The users weren't affected by this patch, only prototype was
- Add a linked list in struct VMA struct to keep track of soft-dirty VMA parts, but will increase the memory usage

WP Asynchronous feature in Userfaultfd and base IOCTL on it

- Userfaultfd uses _PAGE_UFFD_WP PTE flag to Write-Protect the page
- Add Asynchronous WP mode in Userfaultfd UFFD_FEATURE_WP_ASYNC
- Resolve the Page Fault from kernel instead
- A page is considered dirty (written) if it isn't write protected
  - Dirty = !(is_wp(page))

UFFD_FEATURE_WP_UNPOPULATED for Userfaultfd

- Write protection on empty pages isn't recorded
- Add UFFD_FEATURE_WP_UNPOPULATED to remember the write-protection for empty pages by using PTE Markers

PAGEMAP_SCAN IOCTL

- Generic IOCTL to scan the page flags
- Use userfaultfd WP flag in place of soft-dirty flag
- The input of IOCTL is given in struct pm_scan_arg
  - Return compacted data to user in form of ranges
  - Optionally max_pages to find can be specified
  - The scan ending address is returned in walk_end
- Implement the scanning for all memory types:
  - Pages, Huge Pages, HugeTLB and Holes

Supported operations

- Get operation is always performed when output buffer is specified
- PM_SCAN_WP_MATCHING write protects the pages of interest.
- PM_SCAN_CHECK_WPASYNC aborts the operation if non-Async WP pages are found

Filtering support

- category_inverted: PAGE_IS_* categories which values match if 0 instead of 1
- category_mask: Skip pages for which any category doesn't match
- category_anyof_mask: Skip pages for which no category matches
- return_mask: PAGE_IS_* categories that are to be reported in page_region

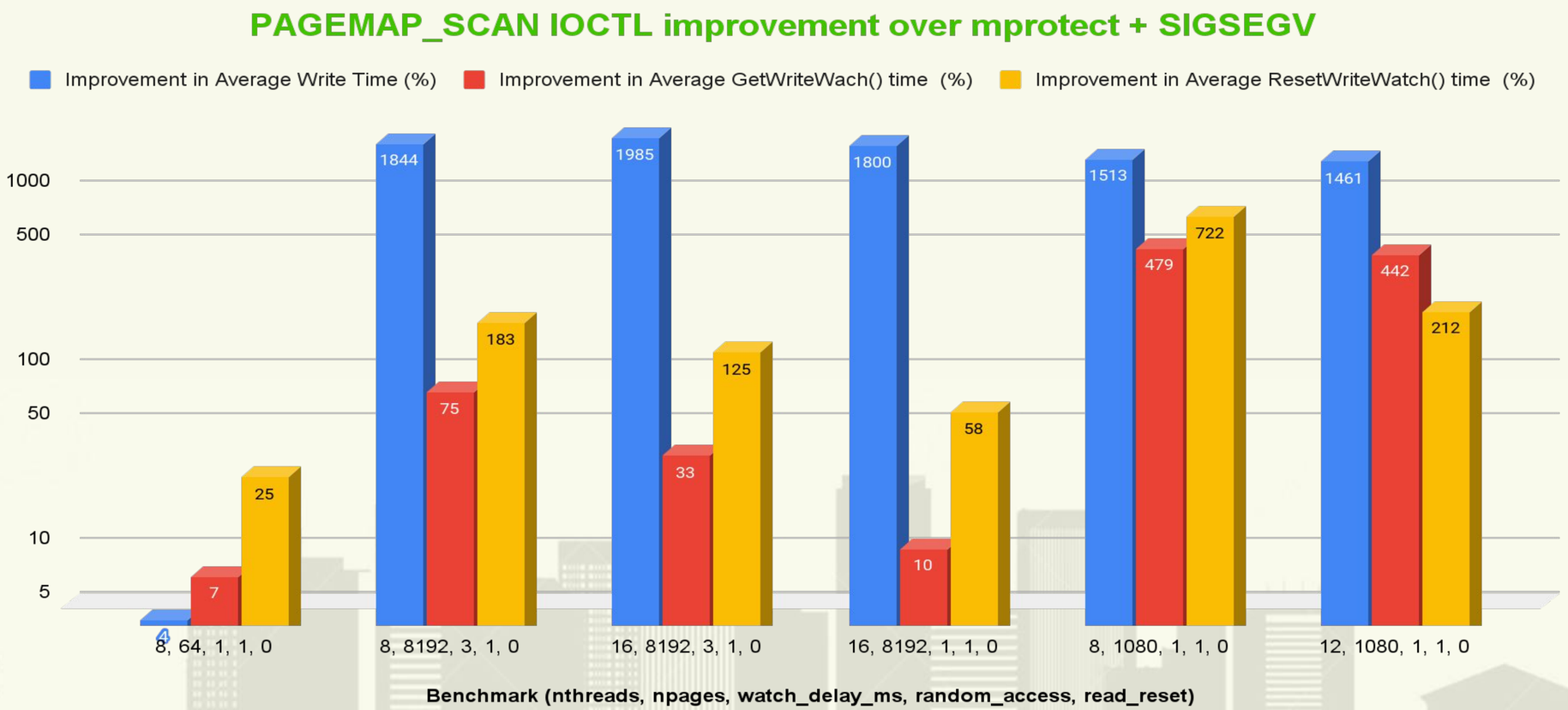
Supported flags

- PAGE_IS_WPALLOWED: Page has async-write-protection enabled
- PAGE_IS_WRITTEN: Page has been written-to
- PAGE_IS_FILE: Page is file backed
- PAGE_IS_PRESENT: Page is present in the memory
- PAGE_IS_SWAPPED: Page is in swapped
- PAGE_IS_PFNZERO: Page has zero PFN
- PAGE_IS_HUGE: Page is THP or Hugetlb backed
- PAGE_IS_SOFT_DIRTY: Page is soft-dirty (WIP)

Performance Improvements

- There were multiple passes to make code more simpler and more performant
- Reduce the internal temporary usage to 12 kB as data cannot be copied to user buffer from inside the mm lock
- Reduce the number of iterations of page walks
- WP the pages and flush TLB only once

Benchmark


PAGEMAP_SCAN IOCTL improvement over mprotect + SIGSEGV
Improvement in Average Write Time (%)
Improvement in Average GetWriteWach() time (%)
Improvement in Average ResetWriteWatch() time (%)
1000
500
100
50
10
5
4
7
25
1844
75
183
1985
33
125
1800
10
58
1513
479
722
1461
442
212
8, 64, 1, 1, 0
8, 8192, 3, 1, 0
16, 8192, 3, 1, 0
16, 8192, 1, 1, 0
8, 1080, 1, 1, 0
12, 1080, 1, 1, 0
Benchmark (nthreads, npages, watch_delay_ms, random_access, read_reset)

PAGEMAP_SCAN in CRIU

- Performance improvements
  - Don't need to revise unneeded pages
  - Batch-mode processing of target pages
- New possibilities for CRIU in unprivileged mode
  - Detect zero pages

    Physican addresses in /proc/pid/pagemap have been hidden to mitigate the rowhammer bug.
  - Track memory change

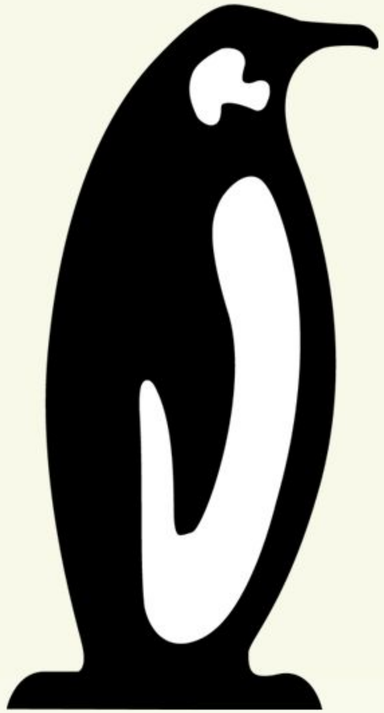
More Use Cases?

Many Thanks-to

- Andrei Vagin
- Andrew Morton
- Michał Mirosław
- Paul Gofman
- Peter Xu
- Et al.

Thanks

Question?



Linux Plumbers Conference

Richmond, Virginia | November 13-15, 2023