



Contribution ID: 192

Type: **not specified**

## Scalability solutions for the `mmap_lock` - Maple Tree and per-VMA locks

*Monday, September 12, 2022 10:30 AM (30 minutes)*

The main issue of `mmap_lock` is its process-wide scale, which prevents handling page faults in one virtual memory area (VMA) of a process when another VMA of the same process is being modified.

The maple tree has been simplifying the way VMAs are stored to avoid multiple updates to the 3 data structures used to keep track of the VMAs.

Latest respin of Speculative Page Faults patchset (<https://lwn.net/ml/linux-mm/20220128131006.67712-1-michel@lespinasse.org/>) was deemed too complex to be accepted and the discussion concluded with a suggestion that “a reader/writer semaphore could be put into the VMA as a sort of range lock”. Per-VMA lock patchset implements this approach.

This talk will cover maple tree and per-VMA lock patchsets as well as the future of Speculative Page Faults patchset and new `mmap_lock` contention findings.

### I agree to abide by the anti-harassment policy

Yes

**Primary author:** BAGHDASARYAN, SUREN

**Presenters:** HOWLETT, Liam (Oracle); BAGHDASARYAN, SUREN; LESPINASSE, Michel (Facebook)

**Session Classification:** Kernel Memory Management MC

**Track Classification:** LPC Microconference: Kernel Memory Management MC