



Contribution ID: 70

Type: **not specified**

Killing the mmap_sem's contention

Tuesday, 10 September 2019 12:00 (45 minutes)

Big systems are becoming more common these days. Having thousands of CPUs is no more a dream and some applications are attempting to spread over all these CPUs by creating threads.

This leads to contention on the mm->mmap_sem which is protecting the memory layout shared by these threads.

There were multiple attempts to get rid of the mmap_sem's contention or the mmap_sem itself, Speculative Page Fault, RangeLock, Scalable Address Spaces Using RCU Balanced Trees...

Unfortunately, these attempts didn't last enough to reach the upstream state. One the reason could be the major impact they are implying on the MM code or that they are only addressing part of the overall picture (SPF).

Last discussions at the LSF/MM summit were not leading to an agreement on a solution (see LWN coverage).

This topic is presenting one of emerging solution which didn't get the time to be proposed at the last LSF/MM. It is based on discussion some folks had at the end of the summit, trying to brainstorm a way to move to a split lock mechanism, as it was done for the PTE locking, removing the mm->page_table_lock.

Currently, this work is still in progress and some deviations on the original design are expected to happen, so kind of split lock is the current option but this may change in the meantime.

This topic is linked to the use of a Maple Tree to replace both the VMA RB tree and the VMA double linked list. Matthew Wilcox and Liam R. Howlett are working on.

I agree to abide by the anti-harassment policy

Yes

Primary authors: Mr GLISSE, Jérôme; Mr DUFOUR, Laurent

Presenters: Mr GLISSE, Jérôme; Mr DUFOUR, Laurent

Session Classification: Kernel Summit Track

Track Classification: Kernel Summit talk