Linux Plumbers Conference 2019

Contribution ID: **104** Type: **not specified**

replacing mmap_sem with finer grained locks

Tuesday 10 September 2019 10:45 (45 minutes)

In the linux kernel, most operations affecting a process's address space are protected by by mmap_sem (a per-process read-write semaphore).

This simple design is increasingly a problem for multi-threaded applications, and often causes threads that operate on separate parts of their address space to end up blocking on each other due to false sharing issues - mmap_sem only supports locking the entire address space at once, so it can't take into consideration that the operations are not overlapping.

I would like to discuss:
1- The sort of blocking issues that are seen today due to the current mmap_sem design;
2- mmap_sem mitigations that have been introduced over time, and have kept the situation bearable but not fundamentally solved the issue;
3- try to discuss from first principles how the MM data structures and locking mechanisms would have to evolve to support finer grained MM locking, and how to progressively migrate the current MM codebase towards such a finer grained MM locking scheme;
4- (hopefully) present early results with a fine grained MM locking prototype.

I agree to abide by the anti-harassment policy

Yes

I confirm that I am already registered for LPC 2019

Primary author: LESPINASSE, Michel (Google)

Presenter: LESPINASSE, Michel (Google)

Session Classification: Kernel Summit Track

Track Classification: Kernel Summit talk