



Contribution ID: 102

Type: **not specified**

Introduce LUF(Lazy Unmap Flush) mechanism

Friday, 20 September 2024 10:45 (45 minutes)

A new mechanism, LUF(Lazy Unmap Flush), defers tlb flush until folios that have been unmapped and freed, eventually get allocated again. It's safe for folios that had been mapped read-only and were unmapped, as long as the contents of the folios don't change while staying in pcp or buddy so we can still read the data through the stale tlb entries.

tlb flush can be deferred when folios get unmapped as long as it guarantees to perform tlb flush needed, before the folios actually become used, of course, only if all the corresponding ptes don't have write permission. Otherwise, the system will get messed up.

To achieve that, for the folios that map only to non-writable tlb entries, prevent tlb flush during unmapping but perform it just before the folios actually become used, out of buddy or pcp.

The result would depend on memory latency and how often reclaim runs, which implies tlb miss overhead and how many times unmapping happens. In my system, the result shows:

1. tlb shutdown interrupts are reduced about 97%.
2. The test program runtime is reduced about 4.5%.

link: <https://lore.kernel.org/lkml/20240531092001.30428-1-byungchul@sk.com/>

Primary author: PARK, Byungchul

Presenter: PARK, Byungchul

Session Classification: LPC Refereed Track

Track Classification: LPC Refereed Track