

Linux
Plumbers
Conference 2022

>> Dublin, Ireland / September 12-14, 2022



The Multi-gen LRU

The ChromeOS team
Google



Current status

- Patchset v14 being tested in linux-next
- Patchset v15 expected to be in v6.1 (LTS)
- 8 downstream kernels carrying the patchset
- 2 additional backports WIP
- 8 server app benchmark results posted
- 2 more server app benchmarks WIP



Linux
Plumbers
Conference 2022

>> Dublin, Ireland / September 12-14, 2022

Next steps

- Make MGLRU the default
- Leverage the page table scanning
- Integrate with eBPF



Make MGLRU the default

Major obstacles

- Free bits in page->flags: need (a few) more
- Performance test coverage: need (a lot) more

Minor concern

- Code health: refactoring needed

Nonissue

- Stability: production ready



Leverage the page table scanning

Unused page tables can cost a fortune

- Reclaiming them may be (relatively) low-hanging fruit
- Discussed during LSFMM 2020
- LWN: “Ways to reclaim unused page-table pages”

Detect empty page tables

- Especially when under memory pressure
- A producer-consumer model may fit in nicely



Leverage the page table scanning

THP internal fragmentation is common

- Accesses to a single base page makes the entire THP seem hot
- Shielding the rest of 511 base pages from page reclaim
- The worst case: those 511 pages have never been used

Detect THP internal fragmentation

- Periodically switch to 512 consecutive PTEs mapping
- Stop protecting fragmented THPs from page reclaim



Integrate with eBPF

The MGLRU framework

- “MGLRU”: simple
- “Framework”: flexible

eBPF programs can generally

- Obtain page access info from MGLRU
- Override the default generation assignment

Specifically they can

- build access heatmaps, supplement `madvise()`, etc.

Demo: access heatmaps





Thank you!