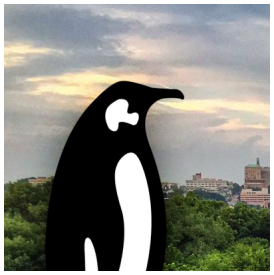

Contribution ID: **179** Type: **not specified**

A move_pages() equivalent for physical memory

Monday, 13 November 2023 17:35 (25 minutes)

CXL-aware job schedulers, memory managers, and userspace tiering solutions depend on page migration syscalls to reallocate resources across nodes. Currently, these calls enable movement of memory associated with a specific PID. Moves can be requested in coarse, process-sized strokes (as with migrate_pages), and on specific virtual pages (via move_pages). However, a number of profiling mechanisms provide information at a systemwide granularity: the IDLE bit is cleared on reads/writes of physical pages, /proc/zoneinfo breaks PFN-space into NUMA nodes, and PEBS/IBS can be configured to track frequency of accesses to physical addresses. Information from these sources facilitates systemwide resource management, but in order to actually *perform* said management, their outputs must be converted back to virtual addresses and re-associated with specific PIDs. Doing this reverse-translation outside of the kernel requires considerable space and compute. It can potentially be avoided if the contents of physical memory can be migrated directly.

We propose a syscall, analogous to move_pages, that migrates the contents of physical pages between NUMA nodes. We present our investigations as to how this call might be integrated with minimal changes to the mm/migrate.c codebase. We also explore a prototype tiering framework that could make use of this call, and its limitations.

Primary authors: Mr PRICE, Gregory (MemVerge Inc); TODOROV, Svetly (MemVerge)

Presenters: Mr PRICE, Gregory (MemVerge Inc); TODOROV, Svetly (MemVerge)

Session Classification: Compute Express Link MC

Track Classification: LPC Microconference: Compute Express Link MC