Contribution ID: **59**   Type: **not specified**

Decoupling ZRAM from a specific backend

Wednesday 11 September 2019 15:00 (45 minutes)

ZRAM is a compressed RAM based block device implementation which has gotten a lot of use recently primarily in the Android world. ZRAM consists of the block device front-end, compressor back-end and memory allocator back-end. Compressor back-end is accessed via a common API, and therefore it is easy with ZRAM to select the particular compression algorithm that fits your special purpose. As opposed to
that, selecting a memory allocator back-end for ZRAM is still not possible because ZRAM is using zsmalloc API directly.

With that said, zsmalloc is not the only kernel allocator for storing compressed objects. There also are zbud (up to 2 objects per page) and z3fold (up to 3 objects per page). Designed to store only integral number of objects per page, these two have deterministic behavior with low I/O latencies. Compression ratio suffers for these two of course – by much for zbud and not so much for z3fold.

Still z3fold might be a better choice as a backend for ZRAM when compression ratio is not as important as keeping latencies low. As a z3fold primary author I keep getting questions when it will be available for use with ZRAM, and keep answering that it has to be a result of a wider consensus. To get closer to this, apart from
zsmalloc / z3fold comparisons, this talk will describe in detail how the existing zpool API should be extended to match ZRAM requirements and whether there is a performance penalty here as this introduces a level of indirection.

I agree to abide by the anti-harassment policy

Yes

I confirm that I am already registered for LPC 2019

Primary author: WOOL, Vitaly

Presenter: WOOL, Vitaly

Session Classification: Kernel Summit Track

Track Classification: Kernel Summit talk