zswap

zswap is a Linux kernel feature that provides a compressed write-back cache for swapped pages, as a form of virtual memory compression. Instead of moving memory pages to a swap device when they are to be swapped out, zswap performs their compression and then stores them into a memory pool dynamically allocated in the system RAM. Later, writeback to the actual swap device is deferred or even completely avoided, resulting in a significantly reduced I/O for Linux systems that require swapping; the tradeoff is the need for additional CPU cycles to perform the compression.[1][2][3]

As a result of reduced I/O, zswap offers advantages to various devices that use flash-based storage, including embedded devices, netbooks and similar low-end hardware devices, as well as to other devices that use solid-state drives (SSDs) for storage. Flash memory has a limited lifespan due to its nature, so avoiding it to be used for providing swap space prevents it from wearing out quickly.[4]

zswap	
Developer(s)	Seth Jennings and others
Written in	C
Operating system	Linux
Type	Linux kernel features
License	GNU GPL
Website	kernel.org (https://kernel.org/)

Contents

Internals
History
Alternatives
See also
References
External links

Internals

zswap is integrated into the rest of Linux kernel's virtual memory subsystem using the API provided by **frontswap**, which is a mechanism of the Linux kernel that abstracts various types of storage that can be used as swap space.[5] As a result, zswap operates as a backend driver for frontswap by providing what is internally visible as a pseudo-RAM device. In other words, the frontswap API makes zswap capable of intercepting memory pages while they are being swapped out, and capable of intercepting page faults for the already swapped pages; the access to those two paths allows zswap to act as a compressed write-back cache for swapped pages.[1][6]

Internally, zswap uses compression modules provided by the Linux kernel's crypto API, which makes it possible, for example, to offload the compression tasks from the main CPU using any of the hardware compression accelerators supported by the Linux kernel. The selection of the desired compression module can be performed dynamically at the boot time through the value of kernel boot parameter `zswap.compressor`; if not specified, it defaults to `deflate` that selects the Lempel–Ziv–Oberhumer (LZO) compression. As of version 3.13 of the Linux kernel, zswap also needs to be explicitly enabled by specifying value `1` for the kernel boot parameter `zswap.enabled`.[1][2][4]

The maximum size of the memory pool used by zswap is configurable through the `sysfs` parameter `max_pool_percent`, which specifies the maximum percentage of total system RAM that can be occupied by the pool. The memory pool is not preallocated to its configured maximum size, and instead grows and shrinks as required. When the configured maximum pool size is reached as the result of performed swapping, or when growing the pool is impossible due to an out-of-memory condition, swapped pages are evicted from the memory pool to a swap device on the least recently used (LRU) basis. This approach makes zswap a true swap cache, as the oldest cached pages are evicted to a swap device once the cache is full, making room for newer swapped pages to be compressed and cached.[1][4][7]

zbud is a special-purpose memory allocator used internally by zswap for storing compressed pages, implemented as a rewrite of the zbud allocator used by the Oracle's zcache,[8] which is another virtual memory compression implementation for the Linux kernel. Internally, zbud works by storing up to two compressed pages ("buddies", hence the allocator name) per physical memory page, which brings both advantages due to easy coalescing and reusing of freed space, and disadvantages due to possible lower memory utilization. However, as a result of its design, zbud cannot allocate more memory space than it would be originally occupied by the uncompressed pages.[3][9]

History

Both zswap and zbud were created by Seth Jennings. The first public announcement was in December 2012, and the development continued until May 2013 at which point the codebase reached its maturity although still having the status of an experimental kernel feature.[10][11]

zswap (together with zbud) was merged into the Linux kernel mainline in kernel version 3.11, which was released on September 2, 2013.[4][12]

Since version 3.15 of the Linux kernel, which was released on June 8, 2014, zswap properly supports multiple swap devices.[13][14]

Alternatives

One of the alternatives to zswap is zram, which provides a similar but still different "swap compressed pages to RAM" mechanism to the Linux kernel.

The main difference is that zram provides a compressed block device using RAM for storing data, which acts as a regular and separate swap device. Using zram requires additional configuration in userspace, using the `mkswap` and `swapon` command-line utilities, so the RAM-based swap device provided by zram is initialized and configured to be used. As a result of its design, zram can provide swap space even if no other swap devices are available, which makes zram more suitable for systems not already providing swap space, such as embedded devices.[15]

In comparison, zswap operates transparently and requires no additional configuration in userspace, and acts as a RAM-based compressed cache for regular swap devices. This provides zswap with an eviction mechanism for less used swapped pages, which zram lacks. Though, as a result of its design, at least one already existing swap device is required for zswap to be used.[15]

See also

- Cache (computing)
- Linux swap
- Swap partitions on SSDs

References

1. Seth Jennings (February 12, 2013). "The zswap compressed swap cache" (https://lwn.net/Articles/537422/). LWN.net. Retrieved January 22, 2014.
2. Jenifer Hopper (December 11, 2012). "New Linux zswap compression functionality" (https://www.ibm.com/developerworks/community/blogs/fe313521-2e95-46f2-817d-44a4f27eba32/entry/new_linux_zswap_compression_functionality7?lang=en). IBM. Retrieved January 31, 2014.
3. Michael Larabel (July 11, 2013). "Zswap Merged Into The Linux 3.11 Kernel" (https://www.phoronix.com/scan.php?page=news_item&px=MTQwODI). Phoronix. Retrieved February 5, 2014.
4. "Linux kernel documentation: Documentation/vm/zswap.txt" (https://www.kernel.org/doc/Documentation/vm/zswap.txt). kernel.org. November 22, 2013. Retrieved January 22, 2014.
5. Dan Magenheimer (April 22, 2010). "Frontswap [PATCH 0/4] (was Transcendent Memory): Overview" (http://article.gmane.org/gmane.linux.kernel.mm/47394). *gmane.org*. Retrieved December 23, 2014.
6. Jonathan Corbet (May 4, 2010). "Cleancache and Frontswap" (https://lwn.net/Articles/386090/). LWN.net. Retrieved March 26, 2014.
7. "Linux kernel source tree: kernel/git/torvalds/linux.git: zswap: add to mm/" (https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=2b2811178e85553405b86e3fe78357b9b95889ce). kernel.org. July 11, 2013. Retrieved February 5, 2014.
8. Dan Magenheimer (March 29, 2012). "Zcache and RAMster (oh, and frontswap too): Overview and some benchmarking" (https://oss.oracle.com/projects/tmem/dist/documentation/presentations/LSFMM12-zcache-final.pdf#page=12) (PDF). *oss.oracle.com*. p. 12. Retrieved August 19, 2015.
9. "Linux kernel source tree: kernel/git/torvalds/linux.git: zbud: add to mm/" (https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=4e2e2770b1529edc5849c86b29a6febe27e2f083). kernel.org. July 11, 2013. Retrieved February 5, 2014.
10. "[PATCH 0/8] zswap: compressed swap caching" (http://article.gmane.org/gmane.linux.kernel/1408711). *gmane.org*. December 11, 2012. Retrieved January 5, 2014.
11. "[PATCHv10 0/4] zswap: compressed swap caching" (http://article.gmane.org/gmane.linux.kernel/1487788). *gmane.org*. May 8, 2013. Retrieved January 5, 2014.
12. "Linux kernel 3.11, Section 9. Zswap: A compressed swap cache" (http://kernelnewbies.org/Linux_3.11#head-e34ed8623c5b96a292064848451d3a7f85faa449). *kernelnewbies.org*. September 2, 2013. Retrieved January 22, 2014.
13. "Linux kernel 3.15, Section 4. Memory management" (http://kernelnewbies.org/Linux_3.15#head-73741bc00ce13369767b5a47dfbcfe07ad68f507). *kernelnewbies.org*. June 8, 2014. Retrieved June 15, 2014.
14. "Linux kernel source tree: kernel/git/torvalds/linux.git: mm/zswap: support multiple swap devices" (https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=60105e1248f571aa3b895cd63bef072ed9d90c77). kernel.org. April 7, 2014. Retrieved June 15, 2014.
15. Dan Magenheimer (April 3, 2013). "In-kernel memory compression" (https://lwn.net/Articles/545244/). LWN.net. Retrieved March 8, 2014.

External links

- zswap (https://www.kernel.org/doc/html/latest/vm/zswap.html) Linux Kernel documentation
- Linux Transparent Memory Compression (https://www.youtube.com/watch?v=jYLzokeRtd0) on YouTube, September 30, 2013, by Seth Jennings
- Zswap – a compressed page add-on for the Linux kswapd (https://web.archive.org/web/20140202233144/http://queen.run.montefiore.ulg.ac.be/~martin/wordpress/wp-content/uploads/2013/02/os-zswap.pdf), University of Liege, March 15, 2013, by Sylvain Martin
- The Compression Cache: Virtual Memory Compression for Handheld Computers (http://www.cs.princeton.edu/~mfreed//docs/6.033/compression.pdf), March 16, 2000, by Michael J. Freedman