Writecache target

The writecache target caches writes on persistent memory or on SSD. It doesn't cache reads because reads are supposed to be cached in page cache in normal RAM.

When the device is constructed, the first sector should be zeroed or the first sector should contain valid superblock from previous invocation.

Constructor parameters:

1. type of the cache device - "p" or "s"
    - p - persistent memory
    - s - SSD
2. the underlying device that will be cached
3. the cache device
4. block size (4096 is recommended; the maximum block size is the page size)
5. the number of optional parameters (the parameters with an argument count as two)

> **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\admin-guide\device-mapper\(linux-master)(Documentation)(admin-guide)(device-mapper)writecache.rst`, line 23)**
>
> Unexpected indentation.

start_sector n (default: 0)
: offset from the start of cache device in 512-byte sectors

high_watermark n (default: 50)
: start writeback when the number of used blocks reach this watermark

low_watermark x (default: 45)
: stop writeback when the number of used blocks drops below this watermark

writeback_jobs n (default: unlimited)
: limit the number of blocks that are in flight during writeback. Setting this value reduces writeback throughput, but it may improve latency of read requests

autocommit_blocks n (default: 64 for pmem, 65536 for ssd)
: when the application writes this amount of blocks without issuing the FLUSH request, the blocks are automatically committed

autocommit_time ms (default: 1000)
: autocommit time in milliseconds. The data is automatically committed if this time passes and no FLUSH request is received

fua (by default on)
: applicable only to persistent memory - use the FUA flag when writing data from persistent memory back to the underlying device

nofua
: applicable only to persistent memory - don't use the FUA flag when writing back data and send the FLUSH request afterwards

    - some underlying devices perform better with fua, some with nofua. The user should test it

cleaner
: when this option is activated (either in the constructor arguments or by a message), the cache will not promote new writes (however, writes to already cached blocks are promoted, to avoid data corruption due to misordered writes) and it will gradually writeback any cached data. The userspace can then monitor the cleaning process with "dmsetup status". When the number of cached blocks drops to zero, userspace can unload the dm-writecache target and replace it with dm-linear or other targets.

max_age n
: specifies the maximum age of a block in milliseconds. If a block is stored in the cache for too long, it will be written to the underlying device and cleaned up.

metadata_only
: only metadata is promoted to the cache. This option improves performance for heavier REQ_META workloads.

pause_writeback n (default: 3000)
: pause writeback if there was some write I/O redirected to the origin volume in the last n milliseconds

Status: 1. error indicator - 0 if there was no error, otherwise error number 2. the number of blocks 3. the number of free blocks 4. the number of blocks under writeback 5. the number of read requests 6. the number of read requests that hit the cache 7. the number of write requests 8. the number of write requests that hit uncommitted block 9. the number of write requests that hit committed block 10. the number of write requests that bypass the cache 11. the number of write requests that are allocated in the cache 12. the number of write requests that are blocked on the freelist 13. the number of flush requests 14. the number of discard requests

Messages:

flush
: Flush the cache device. The message returns successfully if the cache device was flushed without an error

flush_on_suspend
: Flush the cache device on next suspend. Use this message when you are going to remove the cache device. The proper sequence for removing the cache device is:

    1. send the "flush_on_suspend" message
    2. load an inactive table with a linear target that maps to the underlying device
    3. suspend the device
    4. ask for status and verify that there are no errors
    5. resume the device, so that it will use the linear target
    6. the cache device is now inactive and it can be deleted

cleaner
: See above "cleaner" constructor documentation.

clear_stats
: Clear the statistics that are reported on the status line