General Provisions

Cache is used to speed up disk requests by implementing a faster intermediate memory, which stores frequently used data.

Level-1 cache (L1, RAM Cache) uses RAM memory, level-2 cache (L2, flash cache) uses SSD memory.

This article covers the disk cache. Not to be confused with the CPU cache and other types of cache.

Features of L1 and L2 and their implementation in StarWind

Since StarWind allows L1 and L2 caches to use the same algorithms (shared library), most of the entries below apply to both types of cache.

The differences in L1 and L2 functioning are indicated separately.

Cache memory is divided into blocks (currently block size is 64 KB).

Data is evicted from cache using LRU (least recently used) algorithm. If more data needs to be written to the cache, while all the blocks are full, the least recently used blocks are discarded (<https://en.wikipedia.org/wiki/Cache_algorithms>).

Cache block is considered “dirty”, if user data stored in it wasn’t flushed to the disk. In this case, cache memory contains relevant data, while the disk stores an older (out-dated) version of the data.

Cache block is considered “clean”, if data it contains was flushed to the disk (or cache block contains the data read from the disk). In this case both the cache block and the corresponding memory block contain the same data.

Cache block is considered “empty” if unused. At the start of the program all the cache blocks are empty – memory is allocated, but it has no data, and cache blocks are not yet associated with disk blocks.

When the program starts, all the cache blocks are empty, they get filled as the program runs. The “Cache warm up” option is implemented for the lower level cache of the “in-memory” scenario in LSFS: the cache is populated with the data last written to the disk.

Cache Policies

When cache is created, the cache policies are indicated together with other parameters. In V8R5 the cache policies can be changed on the fly for an existing device.

Write-Back Policies:

- works primarily with write I/O;  
- writes new data to the cache, and then confirms the write I/O completion to the host.

If the cache contains clean or empty blocks, WB policies write speed equals RAM write speed.

WB is the preferred mode for L1 cache.

WB is currently blocked for L2 cache (L2 cache is always created in WT mode).

The cache is loaded predominantly during write operations.

Data is cached during write operations only if the cache contains empty blocks or underfilled blocks, previously allocated for the given addresses.

If no new data is written to a dirty cache block within a set time (now 5 seconds by default), its content is copied to the disk. The cache block becomes clean, although it still contains copy of the data.

If all the cache blocks are dirty, the content of the oldest blocks is forcibly copied to the disk, and new data is written to the blocks. Consequently, in continuous write mode, when data is constantly written to different cache blocks, once the WB cache gets exhausted, its performance drops to the parameters of an uncached device: new data is written to a block only after old data is written to the disk.

WB mode boosts performance in case of:

- load fluctuations: when data is written to the cache during load peaks, and then is copied to the disk as the workload declines;  
- continuous rewriting of the same blocks: data is rewritten to the blocks only in the memory, so that multiple writes to the cache correspond to only one disk write.

All the WB dirty cache blocks should be flushed to the disk when the device is removed or the service is stopped. In case of large cache size (gigabytes) this may take extra time. Time estimates for the worst case scenario (all the cache blocks are dirty) can be calculated using the following formula:   
 Flushing time = Cache size / RAM write speed.

In case of an unexpected shutdown (e.g. power outage or service stoppage via Control Panel) the data in the dirty blocks (i.e. not yet copied to the disk) is lost.

For ImageFile HA this results in complete device synchronization when turned on. However this is not critical for LSFS, as it contains technical snapshots where all the data remains intact.

For further clarification, the WB cache data is flushed to the disk in case of:

- no access requests, while the dirty cache blocks are populated with data for more than 5 seconds (this value can be changed via header);  
- new write needs to be made, but all the cache blocks are dirty;  
- device removal or service stoppage.

Write-through Policy

The write-through policy (WT) is used in read operations.

When read from the disk, the data is copied to the cache, so that the next read from the same addresses will be processed directly from the cache, bypassing the disk.

New data is written to the cache and the disk. If a cache block wasn’t mapped with a given address, no new cache block is mapped.

That is, the data is cached when it is read.

When in WT mode, cache blocks are always clean, because new data is written directly to the disk.   
  
Currently WT is default (and only) mode for L2 cache.

In case of device removal or service stoppage, no cache flushing is needed, which enables the system to shut down faster than the WB cache of the same size.

In case of unexpected shutdown program data remains intact.

Conclusion

StarWind uses a two-level cache memory system which enables faster disk requests processing. First level (L1) cache constitutes RAM memory, and is assigned to function in a Write-Back cache policy. Second level (L2) cache constitutes flash SSD memory, and is assigned a Write-Through cache policy. WB policy implemented in the L1 cache speeds up write operations, ensuring the data is copied to the disk in case of an unexpected shutdown, no access requests (over 5 seconds), or a new write. WT policy deployed in the L2 cache allows the data to be written directly to the disk, ensuring the data remains intact in case of an unexpected shutdown.