LIBRBD SETTINGS

See Block Device for additional details.

CACHE SETTINGS

Kernel Caching

The kernel driver for Ceph block devices can use the Linux page cache to improve performance.

The user space implementation of the Ceph block device (i.e., `librbd`) cannot take advantage of the Linux page cache, so it includes its own in-memory caching, called “RBD caching.” RBD caching behaves just like well-behaved hard disk caching. When the OS sends a barrier or a flush request, all dirty data is written to the OSDs. This means that using write-back caching is just as safe as using a well-behaved physical hard disk with a VM that properly sends flushes (i.e. Linux kernel >= 2.6.32). The cache uses a Least Recently Used (LRU) algorithm, and in write-back mode it can coalesce contiguous requests for better throughput.

New in version 0.46.

Ceph supports write-back caching for RBD. To enable it, add `rbd cache = true` to the `[client]` section of your `ceph.conf` file. By default `librbd` does not perform any caching. Writes and reads go directly to the storage cluster, and writes return only when the data is on disk on all replicas. With caching enabled, writes return immediately, unless there are more than `rbd cache max dirty` unflushed bytes. In this case, the write triggers writeback and blocks until enough bytes are flushed.

New in version 0.47.

Ceph supports write-through caching for RBD. You can set the size of the cache, and you can set targets and limits to switch from write-back caching to write through caching. To enable write-through mode, set `rbd cache max dirty` to 0. This means writes return only when the data is on disk on all replicas, but reads may come from the cache. The cache is in memory on the client, and each RBD image has its own. Since the cache is local to the client, there’s no coherency if there are others accessing the image. Running GFS or OCFS on top of RBD will not work with caching enabled.

The `ceph.conf` file settings for RBD should be set in the `[client]` section of your configuration file. The settings include:

`rbd cache`

Description: Enable caching for RADOS Block Device (RBD).
Type: Boolean
Required: No
Default: `true`

`rbd cache size`

Description: The RBD cache size in bytes.
Type: 64-bit Integer
Required: No
Default: `32 MiB`

`rbd cache max dirty`

Description: The `dirty` limit in bytes at which the cache triggers write-back. If `0`, uses write-through caching.
Type: 64-bit Integer
Required: No
Constraint: Must be less than `rbd cache size`.
Default: `24 MiB`

`rbd cache target dirty`

Description: The `dirty target` before the cache begins writing data to the data storage. Does not block writes to the cache.
Type: 64-bit Integer
Required: No
Constraint: Must be less than `rbd cache max dirty`.
Default: `16 MiB`

`rbd cache max dirty age`

Description: The number of seconds dirty data is in the cache before writeback starts.
Type: Float
Required: No
Default: `1.0`

New in version 0.60.

`rbd cache writethrough until flush`

Description: Start out in write-through mode, and switch to write-back after the first flush request is received. Enabling this is a conservative but safe setting in case VMs running on rbd are too old to send flushes, like the virtio driver in Linux before 2.6.32.
Type: Boolean
Required: No
Default: `true`

READ-AHEAD SETTINGS

New in version 0.86.

RBD supports read-ahead/prefetching to optimize small, sequential reads. This should normally be handled by the guest OS in the case of a VM, but boot loaders may not issue efficient reads. Read-ahead is automatically disabled if caching is disabled.

`rbd readahead trigger requests`

Description: Number of sequential read requests necessary to trigger read-ahead.
Type: Integer
Required: No
Default: `10`

`rbd readahead max bytes`

Description: Maximum size of a read-ahead request. If zero, read-ahead is disabled.
Type: 64-bit Integer
Required: No
Default: `512 KiB`

`rbd readahead disable after bytes`

Description: After this many bytes have been read from an RBD image, read-ahead is disabled for that image until it is closed. This allows the guest OS to take over read-ahead once it is booted. If zero, read-ahead stays enabled.
Type: 64-bit Integer
Required: No
Default: `50 MiB`