

Device-Mapper Logging

The device-mapper logging code is used by some of the device-mapper RAID targets to track regions of the disk that are not consistent. A region (or portion of the address space) of the disk may be inconsistent because a RAID stripe is currently being operated on or a machine died while the region was being altered. In the case of mirrors, a region would be considered dirty/inconsistent while you are writing to it because the writes need to be replicated for all the legs of the mirror and may not reach the legs at the same time. Once all writes are complete, the region is considered clean again.

There is a generic logging interface that the device-mapper RAID implementations use to perform logging operations (see `dm_dirty_log_type` in `include/linux/dm-dirty-log.h`). Various different logging implementations are available and provide different capabilities. The list includes:

Type	Files
====	=====
disk	<code>drivers/md/dm-log.c</code>
core	<code>drivers/md/dm-log.c</code>
userspace	<code>drivers/md/dm-log-userspace*</code> <code>include/linux/dm-log-userspace.h</code>

The "disk" log type

This log implementation commits the log state to disk. This way, the logging state survives reboots/crashes.

The "core" log type

This log implementation keeps the log state in memory. The log state will not survive a reboot or crash, but there may be a small boost in performance. This method can also be used if no storage device is available for storing log state.

The "userspace" log type

This log type simply provides a way to export the log API to userspace, so log implementations can be done there. This is done by forwarding most logging requests to userspace, where a daemon receives and processes the request.

The structure used for communication between kernel and userspace are located in `include/linux/dm-log-userspace.h`. Due to the frequency, diversity, and 2-way communication nature of the exchanges between kernel and userspace, 'connector' is used as the interface for communication.

There are currently two userspace log implementations that leverage this framework - "clustered_disk" and "clustered_core". These implementations provide a cluster-coherent log for shared-storage. Device-mapper mirroring can be used in a shared-storage environment when the cluster log implementations are employed.