mdadm Module Documentation

This technical report describes the interface of the mdadm C module, which provides an interface for managing a JBOD (Just a Bunch Of Disks) storage system. The module includes functions for mounting and unmounting the system, a cache system for improving performance, and working with remote JBOD servers.

Code Structure

The mdadm module consists of three C files: mdadm.c, cache.c, net.c and the three files implement the module's functionality as follows:

mdadm.c: This file contains the implementation for the main functionality of the mdadm module, including the functions for mounting and unmounting the disks and reading and writing data from the disks.

cache.c: This file contains the implementation for a cache system to improve the performance of the mdadm module. The cache system is implemented as an LRU (Least Recently Used) cache and is used to cache disk blocks.

net.c: This file contains the implementation for the network communication with the JBOD server. It includes functions for establishing a connection with the JBOD server and sending and receiving data.

Functions and Their Description

mdadm.c must be set up to communicate with the JBOD sever by changing jbod_operation calls to jbod_client_operation. This will ensure that it will connect to the JBOD server where it can send commands. The user can start the JBOD server connection with command [./jbod_server]. Then, the user must run the tester to receive the data, where the following functions are called in the list below.

mdadm_mount()

This function mounts the storage system and returns 1 on success or -1 on failure. It calls the "jbod_client_operation" function with the appropriate command code for mounting the disks and updates the "mounted" global variable. Note that this function must be called first before issuing any other commands, or it will fail. Calling this function the second time without calling mdadm_unmount() in between will also fail.

mdadm_unmount()

This function unmounts the storage system and returns 1 on success or -1 on failure. It calls the "jbod_client_operation" function with the appropriate command code for unmounting the disks and updates the "mounted" global variable. Note that this function should be the last command that is called on JBOD as all other commands after it will fail. Calling this function the second time without calling mdadm_mount() in between will also fail.

mdadm_read()

This function reads a specified number of bytes from the storage system at a specified address and copies the data into the buffer pointed to by "buf". It returns the number of bytes read or -1 on failure. The function first checks if the disk is mounted and if the parameters are valid. It then calculates the disk, block (that is 256 bytes), and block offset corresponding to the specified address and loops through each block until all bytes are read. It calls the "jbod_client_operation" function with the appropriate command codes for seeking to the disk and block and reading the block. It also uses a temporary buffer to store the data read from each block.

mdadm_write()

This function writes a specified number of bytes from the buffer pointed to by "buf" to the storage system at a specified address. It returns the number of bytes written or -1 on failure. The function first checks if the disk is mounted and if the parameters are valid. It then calculates the disk, block (that is 256 bytes), and block offset corresponding to the specified address and loops through each block until all bytes are written. It calls the jbod_client_operation() function with the appropriate command codes for seeking to the disk and block, reading the block, and writing the block. It also uses a temporary buffer to store the block data and copies the data to be written to the current block.

mdadm_cache_create()

This function creates a cache of size num_entries. Dynamically allocate space for num_entires is limited to greater than 2 or less than 4096. If the cache is already enabled or if num_entries is greater than 2 or less than 4096, it returns -1. Otherwise, it returns 1. This is the function the user needs to call if they want to improved performance. Note that calling this function twice without mdadm_cache_destory() will fail. mdadm_cache_destory() must be called at the end to avoid memory leaks.

mdadm_cache_destroy()

This function destroys the cache. It frees the memory allocated for the cache and resets the cache to NULL and cache_size to zero. Note that calling this twice without an mdadm_cache_create() will fail. The num_entries argument also needs to be 2 at minimum and 4096 at most. If the cache is not enabled, it returns -1. Otherwise, it returns 1.

mdadm_connect()

This function attempts to connect to a server with the given IP address ip and port number port. If successful, it sets the global cli_sd variable to the socket descriptor and returns true. Otherwise, it returns false. This function is important as it is called from tester to establish the connection to the server.

mdadm _disconnect()

This function disconnects from the server and resets cli_sd. This function needs to be called to close the connection for the JBOD server.