

# **Online Hierarchical Storage Manager (OHSM)**

fscops@gmail.com

## **Abstract**

Online Hierarchical Storage Management (OHSM) is the first attempt towards an enterprise level open source data storage manager which automatically moves data between high-cost and low-cost storage media. HSM systems exist because high-speed storage devices such as RAM, SSD, rotating disk drive arrays are more expensive (per byte stored) than slower devices, such as optical discs and magnetic tape drives. While it would be ideal to have all data available on high-speed devices all the time, this is prohibitively expensive for many organizations. Instead, HSM systems store the bulk of the enterprise's data on slower devices, and then copy data to faster disk drives when needed. In effect, OHSM turns the fast disk drives into caches for the slower mass storage devices. There would be certain policies that would be set by the data center administrators as to which data can safely be moved to slower devices and which data should stay on the fast devices. Under manual circumstances the data centers suffers from down time and also change in the namespace changes, which eventually leads to breaking of the already running applications. Online here signifies the complete abolishment of the downtime and any changes to the existing namespace for any already running applications.

## **Purpose**

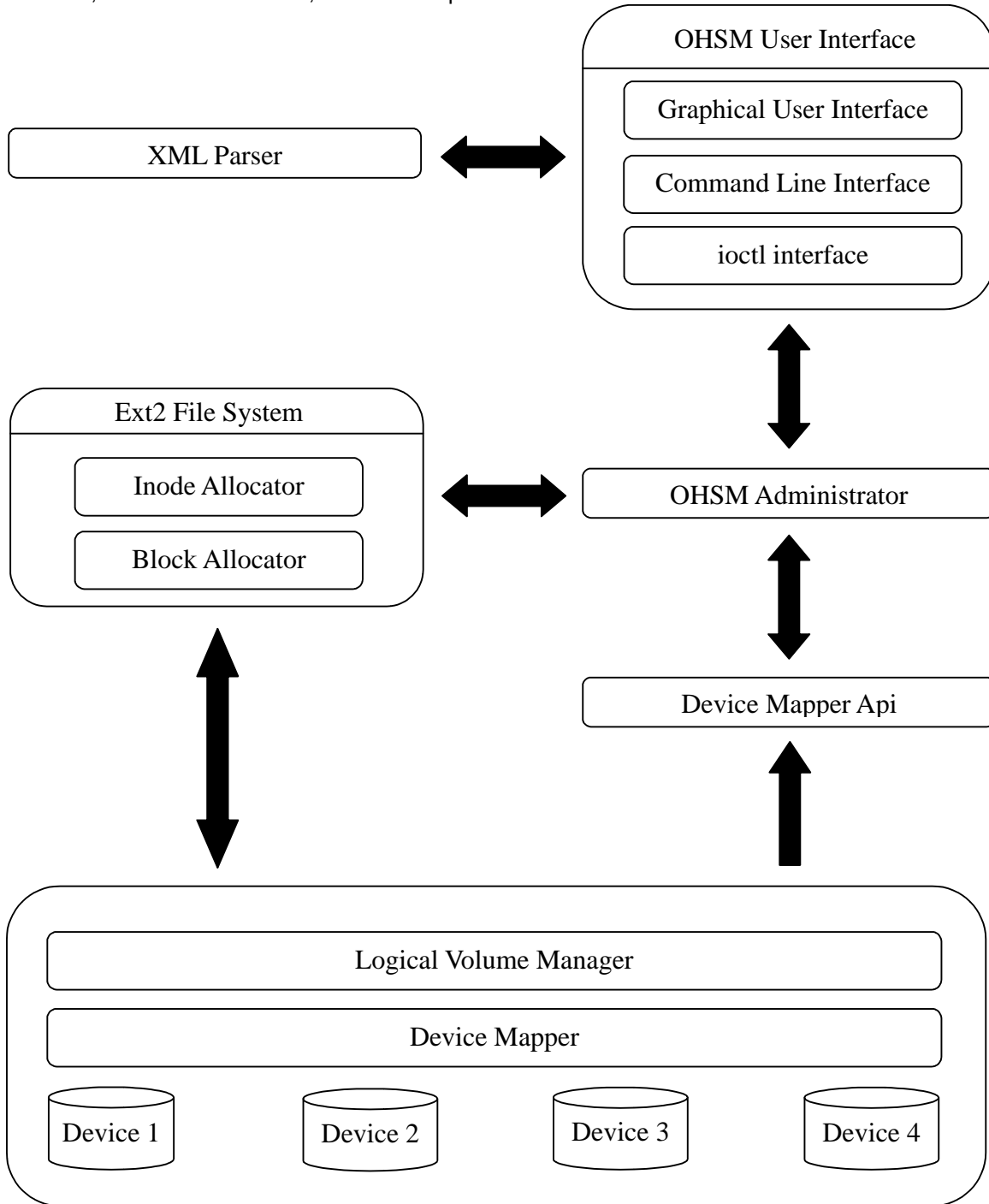
Enterprise level solutions to the storage issues have been provided but they don't benefit normal end user. Also, no work has been done in this area under the roof of open source. As we know that cost of data is getting cheaper and the administrative cost is getting higher day by day, managing storage has become an expensive issue and hitting hard company IT management expense sheets. OHSM provides its users with a fully automated system free of cost and completely open source. OHSM opens a new platform in this area and further works can be carried out to enhance its scope and functionality. It is the first step towards the area of hierarchal storage management in the open source world.

## **Current work in this area**

Significant amount of work is being done in the area of hierarchal storage management and tier based storage. Organizational giants like Hitachi, IBM, Symantec and many more have provided various solutions to ease the complexity of storage management. By classifying and segmenting data into a multi-tiered structure, companies have been creating services and service level policies that support each tier and leverage. IBM offers IBM TotalStorage® whereas Symantec Corporation provides with a similar feature called Dynamic Storage Tiering®, which works with NBU. These products examines the question of how enterprises can effectively exploit available technology and configuration options to substantially reduce the cost of online hierarchal storage for any kind of systems in their data centers without incurring offsetting administrative cost. However, all the related work in this area as of now is completely restricted by licensing, availability and platform specific framework designs. OHSM provides with an open source storage manager which has been solely built for the open source community. This software is freely available and comes under the GNU GPL v3 license.

## Details of the design

OHSM module basically revolves around five modules namely OHSM GUI, OHSM library, OHSM block allocator, OHSM administrator, OHSM dm-api.



Block Diagram – OHSM

- OHSM User Interface provides a graphical user interface and a command line interface to set allocation and relocation policies and also to manage the logical volumes and storage classes. It takes the input from the user and generates a XML file which is stored inside the ohsm driver inside the kernel. XML provides a language for describing the files, tiers, and circumstances under which specific allocations and relocation have to take place. There are sets of policies in xml policy file that interacts with the user space OHSM User interface library. XML policy file contain allocation and relocation policy and storage class information. These XML generated file are then passed to XML parser and the parsed information is passed to OHSM Administrator via OHSM User Interface library through an ioctl interface.
- The XML parser parses the XML file received from OHSM User Interface and extracts all information from it. This information is then passed back to OHSM User Interface OHSM administrator via the help of user space OHSM User Interface Library.
- OHSM User Interface Library concentrates on the ioctl interface. Through these interface you can directly interact with the system and they are responsible for sending the information extracted from XML file to OHSM administrator.
- The OHSM administrator communicates between the user and kernel space functions. It receives the policy file generated by the XML parser and passes the information to the kernel space. These policy files contain information of allocation and relocation of different files and tiers on which they are to be allocated. OHSM administrator maintains internal tables which keep all the information about tier and block range of each device on that particular tier. Also it provides an interface to the OHSM dm-api to get the block ranges and OHSM inode and block allocator for supplying in with block group ranges fetched by the OHSM dm-api.
- The normal block allocation policy would allocate blocks anywhere inside the file system. The goal of OHSM block allocator is to allocate block in a specified place. The distinction is that OHSM block allocator adds restrictions in allocation by specifying the block group range in which file has to be allocated. When a new file is created a function is called for creation of inode. In this function OHSM adds a hook to qualify the file according to file allocation policies. OHSM block allocator module calls another OHSM module function to get the block group range for allocation of file inode. After the file system has the block group range as per the policy to allocate inode for the file it precedes with its normal behavior of creating inode within a specific block range. Once the inode is created we allocate data blocks to it. Similarly, the allocation of data blocks also takes place in the same provided block group range.
- Dm-api is responsible for supplying in with information from device mapper to OHSM administrator. Device Mapper is a core in the Linux kernel which maintains mapped devices (accessible as regular block devices) and their segmented mappings defined in tuples of offset, range, target, and target-specific parameters. Offset and ranges are in units of sectors of 512 bytes. Such tuples are called targets. A list of targets defining segments in the logical address space of a mapped device make up a device mapping table known as dm table which contains complete information of all the devices on which logical volume is created. The device mapper gathers information which is used by OHSM administrator for inode and block allocation. Internally it maintains a linear table which stores all the information such as begin, length and major and minor number which is required to identify the device. This all information is stored in our OHSM table.

## Innovation

The conception of OHSM involves the four major components which are, a user friendly GUI, XML based policy files, a char device interface between user space and kernel space and OHSM block allocator. User friendly GUI allows administrator to easily configure OHSM, forming XML based policy files. Simple text based XML files can be easily parsed to form data structure representing administration information, which is passed to the OHSM char device interface. OHSM coexists with the Ext2 file system and device mapper without altering the normal behavior of the system. OHSM extracts information from device mapper and ext2 for creating its own mapping table. Mapping table contains tiers to file system block group range mappings. OHSM Block allocator uses an optimized allocation strategy to allocate blocks on different tiers. While allocating inodes and data blocks for a file, ext2 qualify the file according to the OHSM policies. Consequently a block group range is referred as per the qualification for allocation. OHSM also provides command line interface for configuration. OHSM offers flexibility in adding and removing storage to any placement class or tiers. OHSM responds to business initiatives with faster storage provisioning and data stability.

## Practical Application

OHSM offers many practical applications.

- **Web Search Engine:** OHSM has its practical application in a web search engine where a regular movement of file is in demand. It offers movement of files on the basis of access, modification, time and many other factors which are required for efficient working of a search engine.
- **Database File System:** OHSM offers movement of data from one tier to another on factors like file type, user, access time, modification which are the key requirement of any database file system.
- **Banking Storage:** In banking organization old transactions are moved to a low priority tier which has low I/O speed and the regular transaction are carried on high priority disk which supports high I/O. OHSM can manage these files or accounts efficiently.
- **Enterprise Storage:** High priority client's information can be stored on a high quality disk and other client information can be stored on lower tier.
- **Mail Accounts:** A regular user's mails can be kept on a faster disk and the less frequent user information can be kept on the lower priority tiers.
- **Video Streaming:** OHSM finds its applications in streaming video over the internet – videos which are regularly accessed can be kept on a high quality tier and rest videos can be accommodated on a low priority tier.

## Components from other project

- ***Libxml2:*** Libxml2 is the XML to C parser and toolkit developed for the Gnome project. It is free software available under the MIT License. We are using this utility to parse the policy files.
- ***GLADE user interface builder:*** Glade is a RAD tool to enable quick & easy development of user interfaces for the GTK+ toolkit and the GNOME desktop environment, released under the GNU GPL License. We have used it for the development of our OHSM graphical user interface.
- ***Logical volume manager of Linux kernel:*** LVM2 is a logical volume manager for the Linux kernel: it manages disk drives and similar mass-storage devices, in particular large ones. We use it to merge the devices.

### Team Member

1. Rishi Bhushan Agrawal
2. Rohit Kumar Sharma
3. Rohit Vashist
4. Sneha Hendre
5. Vineet Agarwal

### Mentor

Mr. Sandeep Kumar Sinha,  
Member Technical Staff,  
RAID,  
NetApp.