

OHSM Allocation FAQs

imreckless@gmail.com

Q1. Can you briefly explain the OHSM allocation mechanism?

Ans. OHSM performs allocation of data blocks for a file according to the active policies set by Administrator on the File System, each time a new file is created, OHSM refers to the active policies on that File System. When a file qualifies a policy, its home tier id is set as specified in the respective policy. Later, whenever data blocks need to be allocated for this file, the home tier id of the inode is referred and blocks are allocated from the specified tier.

Q2. What are the allocation policies supported by OHSM?

Ans. Allocation policies currently supported by OHSM includes:

1. File Type
2. UID
3. GID

Q3. What if the user sets a policy based on UID, which does not exist on the system?

Ans. Actually the method used for defining allocation policies through GUI would never allow a user to do such a thing at the first place because it will only allow to select an UID from a list of existing UIDs. But yes, if the user defines a XML Policy file, then the parser will validate this and will puke an error for the same.

Q4. The allocation policies imply that they are used during the creation of the files, what happens in case of creation of directories?

Ans. It is true that current implementation of allocation policies implies only to regular files and not to directories. OHSM does differentiate the way files and directories are created but does not put any restriction to the creation of directories. So, directories will be handled the usual way by the File System itself. In future, OHSM surely intends to support allocation policies for the creation of directories as well.

Q5. Does OHSM have special handling of device special files?

Ans. The current implementation of OHSM supports only regular files. Once again OHSM puts no restrictions to device special files as well. So, these types of files will be handled as default by the File System.

Q6. What about symlinks and hardlinks?

Ans. Hardlinks refer to the inode of the actual file, and OHSM doesn't move or relocate the inodes of the files. So, hardlinks would not require any kind of special handling to be done. As far as the symlinks are concerned, the same logic implies to them as well. So, overall both hardlinks and symlinks would not require any special treatment from OHSM.

Q7. What happens in case, if at the time of file creation of a file, it qualifies an allocation policy but there is not enough space on the respective tier?

Ans. If a file qualifies for allocation but there is no space in the respective tier, OHSM will notify “no space error” according to the current implementation. The file was intended to be in the specified tier and there was no space left on the tier, hence we raise a flag. If the Administrator wants to handle such situation henceforth, the policies will have to be changed accordingly. For future, OHSM plans to extend itself and introduce the concept of “most preferable placement classes”, wherein if the preferred tier doesn't have enough space then the next preferable tier will be selected. This information would be provided by the Administrator at the time of defining the allocation policies.

Q8. What happens if a file is created according to one of the creation policies, and while writing to it, there are no free blocks on the tier?

Ans. If there is no space left on the respective tier, the file system will report ENOSPC. OHSM has a notification mechanism by which it informs the Administrator that a tier is upto 80% full and a critical warning when a tier is 90% full.

Q9. What happens if a file at the time of creation doesn't qualify any allocation policy?

Ans. If a file does not qualify for any of the allocation policies set on the File System, OHSM will not restrict the allocation to a particular tier. Such files will undergo normal file allocation method of the File System.

Q10. Is the tier information of a file on which it is created persistent on disk? How does OHSM achieve it?

Ans. Yes, the tier information of a file is persistent on disk. OHSM stores this information in the on-disk inode struct of the file.

Q11. OHSM supports allocation policies based on file types, is there any limitation on the same?

Ans. No, there is no limitation on the file type. OHSM will handle all the file types based on extension names specified in the allocation policies.

Q12. What are the other allocation criteria that OHSM intends to support in future?

Ans. Apart from allocation criteria specified in Ans. 2, the future implementation of OHSM will support allocation based on directories, file size and user quota. By directory we mean recursive and non-recursive, both.

Q13. Is there any option to modify the active allocation policies?

Ans. The Administrator cannot modify the active policies. The Administrator will have to rewrite the policy file and reconfigure OHSM with the new allocation policies. In future, OHSM may have support for modification of allocation policies.

Q14. What if an administrator wants to switch OFF any or all of the active allocation policies on a File System?

Ans. Yes, the Administrator can switch OFF any as well as all of OHSM's active policy by simply using the CLI or GUI interface. He can very well switch it ON whenever the need arises.

Q15. What is the method/procedure to apply allocation policies to the FS?

Ans. OHSM provides a user friendly GUI to set/apply allocation policies to the File System. Administrator can also apply policies by writing a XML policy file and applying it through OHSM's command line interface.

Q16. How does OHSM determine the extension of a file? Does it read the file header?

Ans. No, OHSM does not read the file header to determine the file extension. It simply reads the file extension through its directory entry.

Q17. Is there an ioctl to display all policies applied to the File System?

Ans. Yes, an ioctl interface is available to view the current active policies applied to the File System.

For any suggestions/queries, kindly mail the author or fscops@gmail.com.