



HPSS VFS Interface (HPSSFS)

A file system interface into HPSS

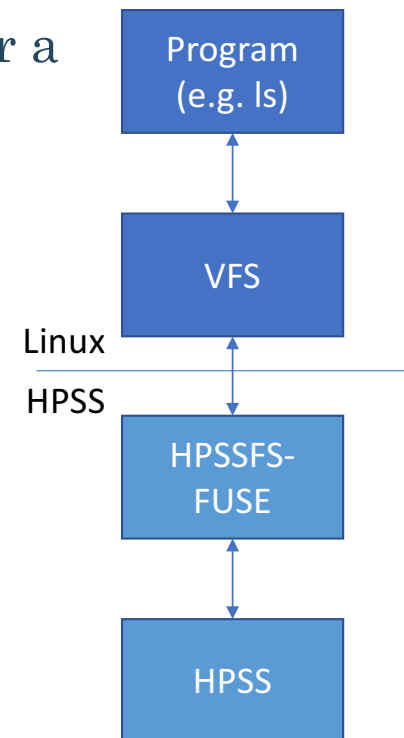


What is HPSSFS?

- HPSSFS is a virtual file system (VFS)
- A VFS is a local or remote file system that is mounted to be accessible to a local system
- The HPSS VFS interface (HPSSFS) enables HPSS to be accessible to local users and applications, just like an NFS mounted file system
- Leverage existing file system applications with HPSS
 - Standard Linux commands such as ls, cat, touch, etc.
 - Shell scripts, custom programs, etc.
- Use standard file and I/O functions with HPSS
 - Standard Linux functions such as open(), read(), etc.
 - POSIX ACLs are supported

Conceptual Architecture

- HPSSFS can provide direct, filesystem-style access to HPSS for a Linux computational cluster
- HPSSFS can enable gateway applications such as Secure FTP, NFS Ganesha, Samba, etc. with HPSS
- Multiple gateways for scalability and sandboxing
- Gateways must provide authentication and security





Appropriate Usage

- Long term data storage
 - While a file system interface is presented, this does not change the workflow or optimal use cases for HPSS
 - Use in a similar manner to other HPSS applications
- Data may be migrated between disk and tape
- Applications must be able to tolerate tape latencies
 - For example, “grep” is dangerous and may generate a huge number of tape mounts
- Linux only



Features

- Linux Extended Attributes
 - Set and retrieve user metadata
 - Set and retrieve HPSS extended metadata
- Automatic checksum generation and verification
 - Endpoint and End to end data integrity
- Data Storage and Movement Flexibility
 - Control how new data is stored in HPSS
- User and Group ID Mapping
 - Map local users and groups to HPSS users and groups
- HPSS IOCTLS
 - Passthrough commands to allow for HPSS specific actions

Linux Extended Attributes

- Allow users or applications, via standard Linux `setfattr` and `getfattr` calls, to associate custom metadata with a file or directory
- This data is in HPSS metadata and will not cause a tape mount

```
fuse# setfattr -n user.color -v blue test
fuse# getfattr -d test
# file: test
user.color="blue"
```



Linux Extended Attributes

- Special system attributes to get HPSS extended attribute metadata
 - system.hpss.level, system.hpss.bitfile, system.hpss.comment, system.hpss.account, and more
- Useful for detecting when data is on disk vs. tape or other HPSS-specific information

```
fuse# getfattr -n system.hpss.level testdisk
```

```
# file: testdisk
```

```
system.hpss.level="0:disk:2048:1048576:1:4194304:(2048:0:[MY0004]);1:tape:nodata"
```



Automatic Checksum Generation and Validation

- Calculate and store file checksums during storage
 - MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA512, crc32, adler32 supported
 - Transparent to the end user
 - Can also be calculated on open and stored if on files without a checksum
- Data Integrity at the User and Server level
 - Validate checksum on opening
 - If checksum mismatch occurs, the application does not get corrupted data back
 - End to End Data Integrity
 - File checksum is checked again upon migration within HPSS
- Checksum metadata is visible to the end user via Extended Attributes



Data Storage and Movement Flexibility

- Data storage can be controlled at the mount point level
 - Class of Service
 - File Family
 - Use large data segments
- Control data movement
 - Specify the data hostname
 - Allow or Disallow SAN3P transfers
 - Allow or Disallow shared memory transfers
 - Enable or disable staging data on open
- Keep small files on disk
 - Files can be “purge locked” to stay available on disk automatically based on their size



User and Group ID Mapping

- Allow local user and group IDs to be mapped to HPSS IDs
- Provides a point of separation so that HPSS IDs need not be extended out to the end client, and vis versa
- Two different modes
 - User Mapping
 - The user who performed the mount will be mapped to a different HPSS user
 - File Mapping
 - A file containing a mapping between one or many local to HPSS user or group IDs
 - The mapping can be between user name or ID



Extensions

- IOCTLS
 - Using the Linux IOCTL interface, an application may make a call to perform HPSS-specific actions against a file
 - Examples: Get the COS, Purge data from the FUSE cache, Purge Lock or Unlock
- `hpssfs_undelete`
 - Utility which undeletes a file in an HPSS trashcan directory
- `fallocate(2)`
 - Allows disk space to be preallocated