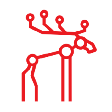
**Moose File System**

**Introduction**

**Moose File System** (**MooseFS**) is an open-source, POSIX-compliant distributed file system developed by Core Technology. MooseFS aims to be fault-tolerant, highly available, highly performing, scalable general-purpose network distributed file system for data centers. Initially proprietary software, it was released to the public as open source on May 30, 2008.

Currently two editions of MooseFS are available:

* MooseFS - released under GPLv2 license,
* MooseFS Professional Edition (MooseFS Pro) - release under proprietary license in binary packages form.

The MooseFS follows similar design principles as Fossil (file system), Google File System, Lustre or Ceph. The file system comprises three components:

* **Metadata server (MDS)** — manages the location (layout) of files, file access and namespace hierarchy. The current version of MooseFS does support multiple metadata servers and automatic failover. Clients only talk to the MDS to retrieve/update a file's layout and attributes; the data itself is transferred directly between clients and chunk servers. The Metadata server is a user-space daemon; the metadata is kept in memory and lazily stored on local disk.
* **Metalogger server** — periodically pulls the metadata from the MDS to store it for backup. Since version 1.6.5, this is an optional feature.
* **Chunk servers (CSS)** — store the data and optionally replicate it among themselves. There can be many of them, though the scalability limit has not been published. The biggest cluster reported so far consists of 160 servers. The Chunk server is also a user-space daemon that relies on the underlying local file system to manage the actual storage.
* **Clients** — talk to both the MDS and CSS. MooseFS clients mount the file system into user-space via FUSE.

**Project Summary**

|  |  |
| --- | --- |
| Website | https://moosefs.com |
| Organization name | Tuxera |
| License |  |
| Open/Proprietary | Open |
| Brief Description | MooseFS Pro is software that allows you to build fast and reliable petabyte file systems on any type of server. It is used worldwide in many bioinformatics laboratories, computer modeling facilities, research institutes, etc. |

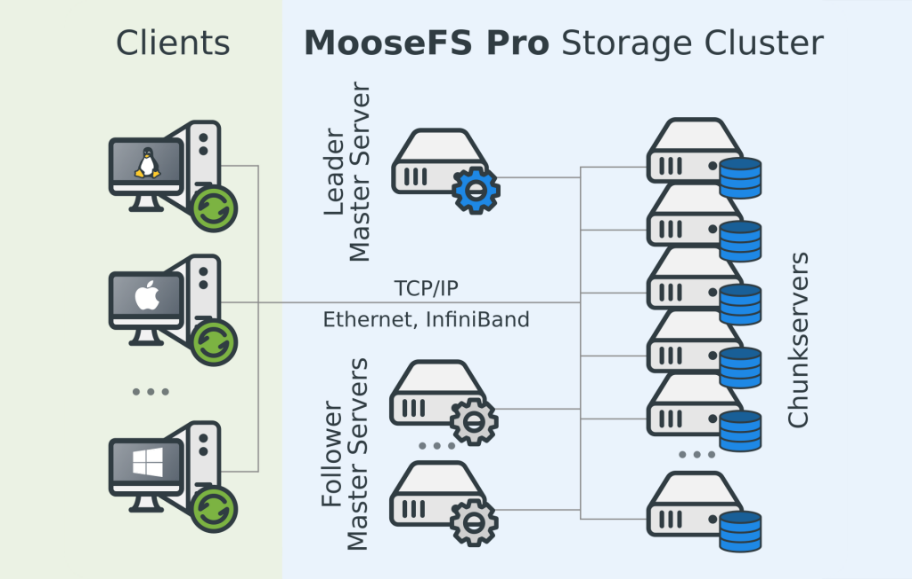
**Project Details:**

**Key Features**

To achieve high reliability and performance MooseFS offers the following features:

* **Fault-tolerance** — MooseFS uses replication, data can be replicated across chunk servers, the replication ratio (N) is set per file/directory. If (N-1) replicas fail the data will still be available. At the moment MooseFS does not offer any other technique for fault tolerance. Fault-tolerance for very big files thus requires vast amount of space - N\*filesize instead of filesize+(N\*stripesize) as would be the case for RAID 4, RAID 5 or RAID 6. Version 4.x PRO of MooseFS implements 8+n Erasure Coding.
* **Striping** — Large files are divided into chunks (up to 64 megabytes) that might be stored on different chunk servers in order to achieve higher aggregate bandwidth.
* **Load balancing** — MooseFS attempts to use storage resources equally, the current algorithm seems to take into account only the consumed space.
* **Security** — Apart from classical POSIX file permissions, since the 1.6 release MooseFS has offered a simple, NFS-like, authentication/authorization.
* **Coherent snapshots** — Quick, low-overhead snapshots.
* **Transparent "trash bin"** — Deleted files are retained for a configurable period of time.
* **Data tiering/storage classes** — Possibility to "label" servers, create label definitions called "Storage Classes" and decide, on which types of servers the data is stored[[10]](https://en.wikipedia.org/wiki/Moose_File_System#cite_note-10)
* **"Project"** quotas support
* POSIX locks, flock locks support

**Architecture**



Typical MooseFS architecture consists of four components.

A MooseFS cluster consists of components of four types: **Master Servers, Chunkservers, Metaloggers (metadata backup              servers) and client machines.**

*Master Servers*

Leader Master Server, Master Server Followers are the machines managing the whole file system, storing metadata for every file and folder.

The leader is always the central point of a cluster. Leader Master Server directs clients to appropriate data servers (Chunkservers) for read/write operations.

It leads all inter-servers data manipulations (replicating, auto-healing, balancing etc.). Master Server neither reads nor sends any file data (chunks).

Each client and data server must know the IP/DNS address of the Master Server. Clients learn from the Master Server addresses of Chunkservers (data servers) for sending or receiving chunks of data.

Master Server Followers are Highly Available online backup copies of Leader Master Server (available with HA configurations).

*Chunkservers*

Data servers (Chunkservers) are machines that store files’ data (chunks). They serve data for clients and synchronize among themselves when necessary.

*Metaloggers*

Metadata backup server(s) (Metaloggers) can be any number of such servers, all of which store metadata changelogs and periodically download the main metadata file.

In MooseFS Community, any Metalogger can be easily set up as a new Master Server in case of main Master Server failure.

*Client machine*

Client computers that use (mount) the filesystem can be any number of such machines contacting Master Servers (to receive and modify file metadata) and Chunkservers (to exchange actual file data) using the MooseFS TCP/IP network protocol.

Linux/FreeBSD/OS X machines use the mfsmount process based on the FUSE mechanism (Filesystem in USErspace) so that MooseFS is available on every operating system supported by FUSE. Also, Windows machines (both desktops and servers) use the native Windows mfs4win driver.

All components communicate with each other using the MooseFS TCP/IP based protocol. The most common network type used for MooseFS clusters is Ethernet but other types supporting TCP/IP stack may be used.

**Current Usage**

MooseFS is a breakthrough concept in the Big Data storage industry. It allows us to combine data storage and data processing in a single unit using commodity hardware thereby providing an extremely high ROI.

Moosefs provide professional services and expert advisory for storage solutions as well as implementation and support for all your operations.

**Technical Details**

* *Redundancy*

All the system components are redundant and in case of a failure, there is an automatic failover mechanism that is transparent to the user.

* *Computation on Nodes*

Support for scheduling computation on data nodes for better overall system TCO by utilizing idle CPU and memory resources. There is no need to buy extra hardware; the same server can be used for storing data as well as for running applications.

* *Atomic Snapshots*

Instantaneous and uninterrupted provisioning of file system at any particular point in time. This feature is ideal for online backup solutions.

* *Redundancy with Erasure Coding*

Ensures data redundancy by using error correction code algorithms with up to 9 parity sums. It saves more RAW space compared to an ordinary data duplication approach.

* *Tiered Storage*

The assignment of different categories of data to various types of storage media to reduce total storage cost. Hot data can be stored on fast SSD disks and infrequently used data can be moved to cheaper, slower mechanical hard disk drives.

* *Rolling Upgrades*

Ability to perform one-node-at-a-time upgrades, hardware replacements and additions, without disruption of service. This feature allows you to maintain hardware platform up-to-date with no downtime.

* *POSIX Compliance*

Allows the user to perform data operations similarly as with the existing file storage systems without the need of making any additional changes. This feature helps in smooth transition from previously used storage systems.

* *Fast Disk Recovery*

In case of hard disk or hardware failure, the system instantly initiates parallel data replication from redundant copies to other available storage resources within the system. This process is much faster than traditional disk rebuild approach. It may take as little as 15 minutes for 1 TB disk recovery.

* *Global Trash*

A virtual, global space for deleted objects, configurable for each file and directory. With the help of this advantageous feature, accidentally deleted data can be easily recovered.

* *Quota Limits*

The system administrator has the flexibility to set limits to restrict the data storage capacity per directory. This feature ensures a stable system and helps in management and storage of crucial data.

* *Native Clients*

Enhanced performance achieved through a dedicated client (mount) components specially designed for Linux, FreeBSD and MacOS systems.

* *Access Control*

Access to files and directories is based on a standard Unix access control model enhanced with standard Access Control Lists.

* *Management Interfaces*

Provides a rich set of administrative tools such as command line based and web-based interfaces, which helps the system administrator to monitor and analyze the system performance.

* *Hardware Independence*

Exceptional performance on almost every hardware platform that runs a POSIX compliant Operating System like Linux, Mac OSX or FreeBSD.

* *Parallelism*

Performs all I/O operations in parallel threads of execution to deliver high performance read/write operations.

* *Ethernet*

1 Gbps / 10 Gbps standard Ethernet-based network can be used for all the communications with the support of LACP configurations.