

ActorFS Data Sheet

Reactive Object File System for Big Data

ActorFS is a reactive cluster object file system for big data to store files and objects. Peer to peer for elasticity, shared-nothing for reliability, reactive to be asynchronous and non-blocking to have better use of multi-core architectures. With fully distributed asynchronous copy-on-write indexer to support massive amount of small file and object creation in-memory, over-disk and with-SSD and to leveraging redundancies in cluster devices with parallel journaling.

Fusion of File & Object Concept

ActorFS fuses file and object concept by integrating three different layer of storage on a single system that are Page System, Object System and File System. Fusion of highly coupled data layers makes the system more efficient on performance optimization.

Store Files: 1 Byte to 1 Peta Byte

ActorFS is designed to scale on large number of files and encourages to create file whenever needed even for 1-Byte of data. The storage engine is efficient enough to handle large files even petabytes for a single file, object sets, and object collection on top of the file system. To differentiate from No-SQL specially key-value stores, ActorFS gives similar functionality but at the lowest possible level of data storage to boost performance. Benchmarks show 420K object insert and 600K in-memory read of different files in a single node deployment on a mid level laptop and is scalable linearly on cluster deployment.

Eliminate Index Complexities

ActorFS encourage to create files not to create structures to find records in a file. Traditionally database systems where to solve this problem by adding primary or secondary indices to solve problem of intra-file or table search. ActorFS uses the main indexer to do this by allowing user to create large number of files each may having only one record.

Object Serialization

ActorFS is capable to migrate objects, codes and states throughout the cluster. The API is capable to take snapshot of serialize able objects to have them on different machines.

Console

The file system uses a client console that is on top of Scala interpreter. The console allows to find, add , remove or list objects and collections in the file system. ActorFS is equipped with domain specific language named FSL. Using FSL users and developer may program the file system to do complex tasks.

Stream Encryption & Compression Engine

60x Speedup of Encryption without CPU overhead is achieved using CUDA integration of the core file system design. The Compression Engine uses CUDA to enhance throughput of compression to 0.8 Gbps without CPU overhead.

Perform/Transform Model

Simplifies distributed Processing, using push based delivery of think small and execute in the large model. With ActorFS the user may submit data actors in-the cluster to deliver data to them in-memory or over-disk.

Semi-Ordered Object Streaming

ActorFS uses enhanced mechanisms to manage multiuser concurrency. ActorFS encourages using object mode operations to support semi-ordered data retrieval.

Objects & JSON

ActorFS supports JSON by converting it to text/string objects. However ActorFS encourage using Actual objects without unnecessary conversion between different formats that creates a lot of problems with binary data and representation. This mechanism enables fast data retrieval.

Responsive on Parallel Read

ActorFS uses pages instead of large blocks for addressing that makes system more responsive by utilizing more disks.

Interfaces & Integration

Hadoop & Spark Integration

Giving variety of interfaces that are POSIX, HCFS, REST, AKKA interface and Scala/Java API