MarFS Metadata Scaling

David Bonnie, Hsing-Bung Chen, Gary Grider, Jeffrey Inman, Brett Kettering, William Vining

Overview

MarFS is a near-POSIX file system interface on top of Cloud-Style erasure and objects. Erasure coding allows you to build very reliable storage systems out of unreliable and inexpensive disk technologies while objects enable extreme data scaling. MarFS is appealing to large data sites because they have large numbers of legacy applications that rely on POSIX file system semantics (files, folders, ownership, etc.), but that require the reliability of erasure and the scalability of objects.

Our use of Parallel File Systems is to store large sets of data for weeks to months at extreme scales, order 1 TB/sec. Our parallel archives are used to store data forever at very modest speeds, order 10s of GB/sec. MarFS was designed to provide a place to store very large sets, order PBs, with aggregates of Exabytes of data for years with speeds order 100s of GB/sec. The supercomputers generating the data MarFS ultimately stores are in the millions of cores and PBs of memory now and are expected to grow to a billion cores and 10s of PBs of memory. 1 file per process applications could produce billions of files that a user may want to reside in the same directory. Further as we push to add value to the data we store we expect file-oriented metadata to grow by perhaps orders of magnitude. The desire is for MarFS to handle billions of file metadata entries in a single directory and 10s of trillions of metadata entries in aggregate.

The work described in this WIP (Work in Progress) report covers the work to enhance MarFS to handle this extreme scale metadata workload described above. To do this we are enhancing MarFS from scaling its file and directory metadata together to scaling file metadata and directory metadata independently, while maintaining the ability to continue to scale the file data independently.

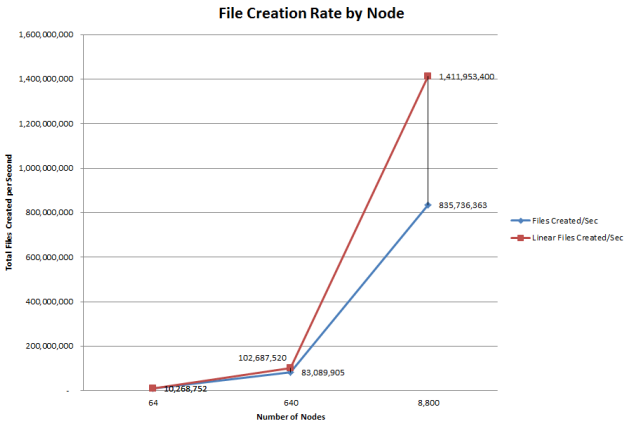
Basic Architecture

The basic components are:
drMDS: The directory metadata server. There is always one (1) of these and it is rank 1 on the first node in the allocation. Its only job is to make new directories and broadcast the directory's inode out to the fsMDSc's.
fsMDSc: The file system metadata server for collectives. This is the server that handles its sharded part of the distributed file metadata, but only receives commands that are broadcast to all file system metadata servers.
fsMDSp: The file system metadata server for p2p. This is the server that handles its sharded part of the distributed file metadata, but only receives commands that are intended for it to handle its part.
Client: The processes used to create files or do other file system operations.

Early experimentation showed us that it was better to distribute the clients and servers over the nodes such that a node's cores are 1/2 Clients, 1/4 fsMDSp's, and 1/4 fsMDSc's.

File Creation Results

Thus far, the only scaling tests we have completed are parallel file creation into the same logical sharded directory. Using the ACES system, Cielo, a small scaling experiment was done to see how many files we could create per second. Each Cielo node has sixteen (16) cores. There were eight (8) Clients, four (4) fsMDSc's, and four (4) fsMDSp's. The largest number of total files created into a single logical directory was over 900 Billion (almost a trillion files in one logical directory) at a rate of over 800 million file creations per second.



Future Work

In the future we intend to do more extensive parameter studies for creates, stats, readdir (sequential), and readdir (parallel). We'll do these with our demo application using straight MPI/POSIX file operations and MarFS file operations. We intend is to see what level of overhead MarFS imposes over straight MPI/POSIX.