Solving the Challenge of Lots of Small Files (LOSF)

Why Metadata Management Matters

Challenges

- The scale and performance customers expect from applications that read and write lots of small files exceeds what legacy architectures can provide
- Scaling and parallelizing metadata operations as is inherently more complex than scaling the performance of data IO

Solution

- The WEKA® Data Platform patented data layout and virtual metadata servers distribute and parallelize all metadata and data across the cluster for incredibly low latency and high performance no matter the file size or number

Benefits

- Increased time-to-value for EDA, AI/ML, and a variety of custom HPC applications
- Simplified and easier to manage data infrastructure on-premises
- Ability to move small file-intensive workloads to the cloud and maintain performance

Becoming a data-driven company not only requires you to expand your data storage capacity, but also to adapt to significant changes in data types and data size. One of the challenges is the rise and growing importance of applications that use lots of small files (LOSF). Despite the unassuming acronym, these can be quite a challenge for your legacy data infrastructure.

Academia and the industry generally consider “lots” to be a million files or more and small files to be less than 1MB in size. But in the era of next-generation workloads, LOSF applications can often produce not just millions, but billions of small files, and mix them among large files. And to make matters even worse, 1MB size may be on the large end. Many of the files in these workloads are only tens of kilobytes in size or smaller. LOSF applications are common in social networking sites, electronic design automation, high-performance computing, and natural language processing among others. LOSF introduces new and significant challenges in metadata management, access performance, and storage efficiency. LOSF workloads are also particularly challenging to move to the cloud.

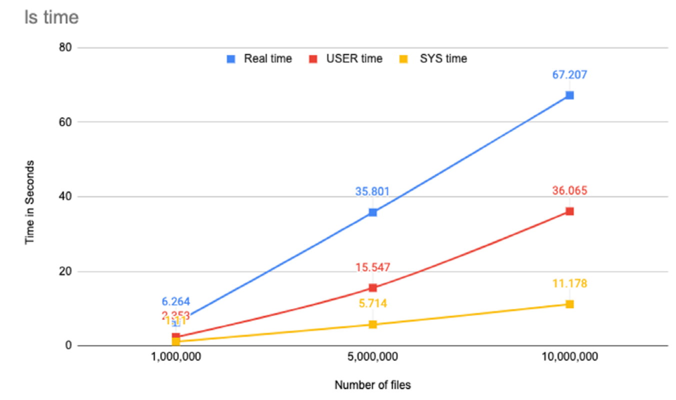
The Root of the Problem

Current file systems that originally focused on the HPC space, such as Lustre, GlusterFS, GPFS, and HDFS were architected for a problem of a different era and not for low-latency access to small files. Originally designed for slower hard disk storage of the past, they were architected to deliver high aggregate bandwidth for large files, including how they dealt with metadata management, data layout, striping design, and cache management. Because of this, the performance and storage efficiency of these file systems will be significantly reduced for massive small file applications.

Let’s break down the root of the issue: metadata management. In a filesystem, you have the data itself represented as files. In addition, there is a required set of pointers to the files that describe their location on whatever media is

Many Files in a Directory

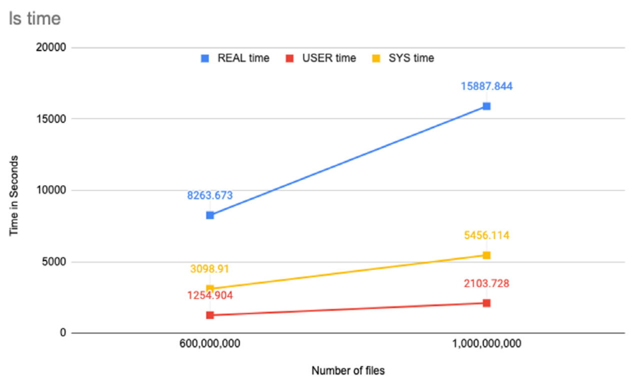
1 million, 5 million, 10 million files in a single directory



being used (the directory structure), and other attributes such as when the file was created, when it was last modified, and more. When a file needs to be read or written, the client (server or computer) asking for it first makes a request to the storage to look up or find the appropriate location, then it opens the location, performs its reads and/or writes, and then closes the file. This means for each data movement, there will be multiple metadata operations. This metadata needs to be stored and protected within the system. In a traditional enterprise storage system, metadata is managed within the controller of a set of disks. In some cases, the metadata can be managed by having copies of it and passing it across multiple controllers or even somewhat distributed across a limited set of controllers. In the case of many HPC file systems, there are dedicated metadata servers outside of the storage to handle the number of metadata operations the filesystem needs to function.

Many Files Many Directories

600 million files across 10,001 directories – 1 billion files across 20,002 directories



- ls 600 million files in 2 hours and 30 minutes
- ls 1 billion files in 4 hours and 41 minutes

At the small scale and limited performance of past devices like HDDs, this metadata was somewhat manageable and strategies that could handle it were appropriate at the time. Today, with environments reaching 10s of millions or even billions of files along with the performance customers expect from these new types of LOSF workloads, the strategies of the past have begun to fail. The metadata design of traditional enterprise storage can't scale and winds up swapping metadata from disk to memory at a tremendous rate. And the HPC file systems with dedicated metadata servers are complex and require sizing exercises and tuning for each file system under its control in order to attempt to meet specific performance criteria.

Cloud to the Rescue?

As more LOSF workloads move to the cloud, surely existing cloud storage offerings can tame this monster, right? While most object store implementations in the cloud can handle the scale of billions of files, the performance is far too slow for any of the next-generation workloads. The enterprise storage vendors that have moved to the cloud have just replicated their metadata shortcomings and in some cases have made it worse by reducing available capacities or decreasing the number of controllers in a cloud deployment. In the cloud, HPC-style storage is typically a managed service which means that while it is relatively simple to deploy, the ability to tune it to meet the LOSF application needs is lost.

As LOSF-style and other next-generation at-scale applications become more mainstream and pervasive in the enterprise, the demands for performance regardless of location will increase. Tradeoffs for operational ease vs. performance and scaling have an impact on how quickly you can achieve time-to-value for EDA, AI/ML, and a variety of custom HPC applications. These tradeoffs will be tolerated less and less moving forward.

WEKA Solves the LOSF Problem On-Premises and in the Cloud

In a world where data workflows and pipelines are consolidating onto single storage systems and increasingly becoming a blend of LOSF and large data files at the same time, having the ability to handle ALL the IO regardless of size is becoming critical. The WEKA Data PlatformTM takes a fresh approach to this problem by creating virtual metadata servers that scale on the fly with every server that is added to the cluster.

WEKA runs multiple virtual metadata services, across the CPU's in a cluster.. So even in a small environment with just six nodes, there are thousands of virtual metadata servers that are running across the cluster. The metadata plane is sharded into thousands of small "buckets" so that each virtual metadata server handles a small part of the namespace. As clients access different files, they access different metadata servers. And, every 8K is handled by a different virtual metadata server so even if multiple clients access a single file in different block ranges, the load is distributed. Load balancing across nodes prevents hot spots from forming and spreads the load across all metadata buckets in the system. The metadata servers journal to the WEKA storage to provide resiliency. The metadata management is calculation-based which eliminates memory-based look-up tables.

Along with WEKA's patented data layout algorithms which distribute and parallelize all metadata and data across the cluster in small 4k chunks, this creates incredibly low latency and high performance whether the IO size is small, large, or a mixture of both. Because the WEKA Data Platform is software-defined, the exact same technology is used on-premises or in the cloud, providing you the same benefits and user experience..

Different Approaches to Handling Metadata

Appliance Filesystem

Traditional NAS/SAN appliances based on High availability pairs of controllers have Metadata limited to the domain of the HA pair. Because of this, as the filesystem gets larger, the amount of memory needed to cache metadata information to achieve high performance gets larger and larger. It's not unusual for high-end versions of controllers to have 512GB-1TB of memory to hold the metadata. If the HA pairs form a scale out cluster, there is usually very limited sharing of metadata across the cluster. In many cases, files cannot cross boundaries of the HA pair and so are limited to the metadata available within the HA pair only, even if the filesystem scales across the cluster.

Parallel File System

Legacy parallel file systems originally stored metadata on separate, dedicated metadata servers (MDS) that manages the entire namespace. One set of MDS servers services the entire cluster, no matter the size. This is efficient but can also be a bottleneck as the filesystem scales. To maximize performance, MDS tried to keep as much of the metadata in memory, requiring a large and expensive cache for large filesystems. And when memory can no longer hold all the metadata, it needs to be swapped to and from disk to memory at a tremendous rate with resulting latency. Every file IO requires multiple metadata operations to locate the file. So in a scenario when the filesystem load increases with lots of small files, the load on the single MDS significantly increases as it gets bombarded with metadata requests. And as the load increases the performance of MDS slows down which slows down the performance of the entire file system. This also presents a single point of failure and requires a failover MDS to take over in case the primary has a planned or unplanned outage.

Distributed Parallel File System

With newer distributed Parallel File systems, you can have metadata management, with multiple active MDSs manage a file system's namespace. This way, the workload is shared across several servers to improve performance and scalability. Memory requirements are still high for large filesystems, but spread over more servers to allow for less overall RAM per server.Because no single MDS manages the namespace, there is coordination between the distributed servers to maintain consistency. A single metadata operation may update several different MDSs, and is usually checksummed for all operations so in the case of transmission or other errors, the MDS update can be validated before being committed.