Facebook Warm Storage – next generation storage for Data Warehouse in Hadoop ecosystem

[Authors]

Introduction Very few companies run 24/7 clusters at the scale like Facebook using Hadoop/HDFS. As scale increases HDFS shows its limitations in scalability, performance and availability. HDFS was first designed 10+ years ago based on the assumptions that were valid in 2005. Over the years, numerous significant improvements have been made to the system. However storage, network and compute technology changed at different ratios during last decade. In this paper we describe Warm Storage - grounds up solution that exploits technology inflection points that exist in 2015 and we believe will continue into the future. We describe architecture and some of the implementation details of the novel and more efficient and available storage system optimized for DW workloads.

What had changed since birth of HDFS In 2005 when HDFS was created disks had about 0.5TB capacity and the best data retrieval rates ranged at ~100MB/s. Intra DC network was just switching to 1 Gbps links and TOR switches had ~1:40 oversubscription. 10 years later while disks have 4-8TB or even 10TB of storage capacity bulk data retrieval rates almost haven't changed. However 40-100 Gbps per port data center networks are common and CLOS style network topology allows creation of very large data center deployments where each host can utilize full (or a very significant fraction) of NIC bandwidth. With such disk/network speed ratio data locality optimizations that were key in designs of circa 2005 when HDFS was created are no longer critical. Indeed, by giving up on the attempt to achieve network locality we can create totally new solutions of increasing availability and performance for DW storage. One of the most significant design contributions in Warm Storage is dis aggregation of the storage and the compute hosts where all storage access is done remotely over the network. This separation also allows to create a system with much more intentional control of the hot spots and tighter high percentile latency bounds on the system responses. Also separation is an essential instrument in the life cycle and the economy of the data center from the deployment to decommissioning stages.

HDFS problems at scale Metadata (managed by service called NameNode in HDFS) scaling issues are one of the most classic problems with multiple proposed solutions by many contributors. In Warm Storage we approach this problem from the pragmatic industry point of view and create a scalable metadata layer over the block pool designed to handle number of files and blocks measured in trillions. We approach this by reusing exiting DB technologies specifically MySQL and other scalable NoSQL solutions at Facebook.

Improving HDFS availability In very large DW deployments design choice in HDFS start becoming a limiting factor on the overall system availability when access to each file is critical. In Warm Storage we are using synchronous Reed-Solomon encoding and we use tradeoffs between storage, network and compute not just to increase durability of the data, but also to achieve and control overall system availability. Compared to HDFS we reduce probability of all files being unavailable by multiple orders of magnitude.

Data storage efficiency While multiple proposed and implemented contributions based on HDFS do address 3x replication cost by doing asynchronous Reed-Solomon encoding, operationally such implementations often are not as effective and do not achieve theoretically expected reduction in the storage costs. In Warm Storage we use synchronous RS encoding and have a novel contribution about improving RS compute efficiency via code generation.

Operational efficiency in HDFS Only when operating at the scale HDFS operational overhead and load on people performing troubleshooting and investigations becomes obvious. At the core of HDFS a self-organizing architecture of Data Nodes (hosts performing blob data storage) is effective, but operationally very costly to manage and to troubleshoot. In Warm Storage we approach this by creating an introspectable storage system where we can always understand and act effectively to recover and repair malfunctioning parts. Our contribution is about keeping this information at the manageable level so that even clusters that are deployed at Facebook scale can be efficiently operated.

Predictable service and multi-tenant isolation HDFS isn't designed to deliver predictable response time for read or write operation. It also mixes internal system upkeep operations with the requests from customers and essentially is designed as best effort storage service. This leaves a possibility where any of the requests to HDFS can bring system to a starvation point or enable very unfair sharing. In Warm Storage we have a design where we can deliver tighter bounds on high percentile latency for read/write operations by exploiting Reed-Solomon encoding and quorum repairs. We also do proactively management and separation of internal system upkeep workload from customer requests as well as requests of different customers. This allows us to achieve significantly better latency and even throughput.