DSS Gen2, High Bandwidth AI Training Storage Platform

Ronald Lee
Samsung/MSL
San Jose, USA
r2.lee@samsung.com

Mayank Saxena
Samsung/MSL
San Jose, USA
mayank.s4@samsung.com

Somnath Roy
Samsung/MSL
San Jose, USA
som.roy@samsung.com

Harsh Roogi
Samsung/MSL
San Jose, USA
h.roogi@samsung.com

Aaron Lee
Samsung/MSL
San Jose, USA
a.lee03@samsung.com

Abstract

Increasingly, as AI technology evolves into more sophisticated applications, training dataset sizes continue to grow exponentially. In order to scale storage and network infrastructure commensurately to deliver the data required and avoid unbearable training cycle times, there is a need for a new storage concept and innovative solution to address this technology gap. DSS Gen2 and beyond provides a potential next generation architecture and solution to alleviate this bottleneck.

Another issue that is major impediment to data center scaling is power utilization. And since one of the key storage consumer is data storage, DSS technology not only plan to optimize server power utilization but also partner and leverage Samsung's new high capacity SSDs which has one of the highest density in the world.

I. Introduction

An ideal storage solution would be to enable full bandwidth utilization of existing/available network, server and storage technologies to deliver high bandwidth data to AI frameworks. In addition, allowing this platform to scale linearly as additional network, server and storage components are added to the infrastructure is a must. This is the core concept behind DSS Next Generation, using Object Storage technology + Protocol to simplify network transactions and then optimizing data transfers from SSDs through server system to network, where data is directed onto GPU memories for immediate processing. As faster network, server and storage technologies comes into existence, DSS platform aims to take full advantage.

II. DSS Gen2 Architecture Concept

The second generation DSS platform extends the architecture by enabling direct data transfers from storage to GPU memories. DSS is unique in that it utilizes proprietary S3 extended protocol to allow Object Storage systems such as DSS to be able to leverage RDMA operations. In this architecture, key value data is transferred directly from SSD device to GPU memory without incurring any other overheads. Since DSS stores S3 object data as erasure coded Key Value chunks on individual Node+SSDs, DSS Target storage engine only requires Keys and the corresponding destination GPU RDMA addresses for these erasure coded chunks to process the respective data transfers.

In addition, unlike file based storage solutions such as Lustre and BeeGFS, where if one or more of the scale out storage nodes fail, erasure code recovery is handled by the clients. This unique architecture relocates the data recovery handling to client side instead of burdening storage servers. This is an important architectural change since erasure code recovery can cause variety of issues to the storage cluster including severe performance degradation, reduced ability to meet quality of service, significant increase of system and network traffic, and additional CPU/Resource usage of involved servers and storage devices. Although, this might look like we have effectively just moved the problem to the client side, we would argue that this architecture is much more scalable and provides for predictable environment.

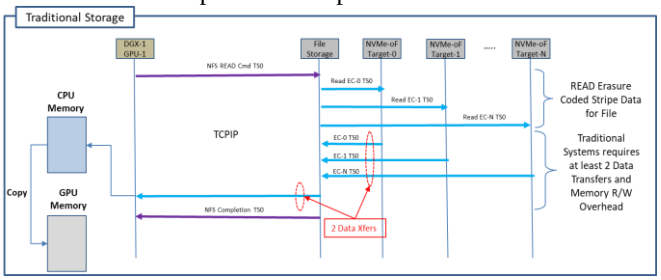

Figure 1: Requires almost 2x Data Movements

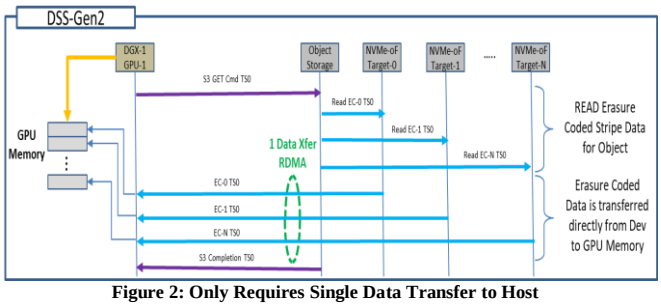

Figure 2: Only Requires Single Data Transfer to Host

In this architecture, the erasure code data recovery is handled by the client/network (note: it is possible to embed erasure code recovery into the network component for lower latency and reduce client overheads), this offloads the storage cluster from having to rebuild the failed data chunks and thus eliminates significant load from the storage cluster. Rebuild operations requires data to be transferred between nodes on the storage cluster and then the resultant data to the client. This causes at least 2x inflation of network traffic especially when you are trying to achieve Terabytes/sec storage cluster bandwidth. In addition, the storage nodes that are handling the rebuild operations will be burdened

with additional processing and thus less likely to meet any kind of existing quality of service guarantees.

DSS Gen2 architecture concept allows the storage cluster to continue operating even during a node or device failure as if everything was normal except that failed devices obviously will not be able to transfer any data to the client. However, the storage cluster can identify which erasure coded data chunks have failed and notify the client what is missing in the returned response. The S3 protocol extension also contains provisions for rebuild operations by allowing information in the returned response to indicate methods to rebuild the missing chunks. Thus the client has the information and knowledge required to rebuild any erasure coded chunks that are missing. It is possible to handle this rebuild operation in the GPU, CPU or offload this into the network using some sort of intelligent NIC/Switch Port. Obviously, using specialized hardware such as GPU or Network Component will be desirable to reduce load on Client CPU.

In this new architecture, storage cluster bandwidth is preserved during node or device failure where remaining nodes and devices are still operating at full speeds. Essentially, failures do not impact other clients that are sharing the storage cluster and quality of service guarantees are not impacted since all other traffic is running at full bandwidth. This is highly important, especially in a shared AI Training or similar environment where customers might have multiple concurrent data/model parallel training operations on-going simultaneously. Although, there will be an additional burden on the client nodes, this is much more predictable environment and can be overcome by having faster client servers/CPU, GPU and related hardware resources.

Additionally, when the failed nodes and devices are replaced, it is imperative that the storage cluster minimizes performance impact during reconstruction of this data. Fortunately, this can be mitigated by controlling the pace of data reconstruction and possibly doing this during periods of reduced storage cluster usage.

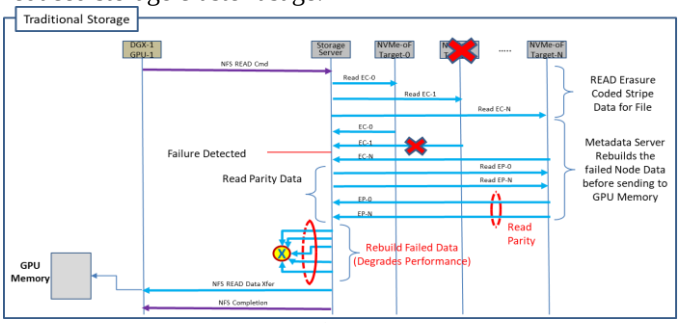

Figure 3: Rebuild is performed on Storage Server

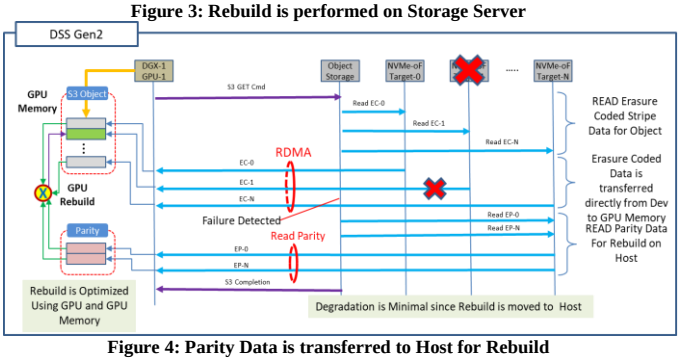

Figure 4: Parity Data is transferred to Host for Rebuild

III. S3 PROTOCOL WITH RDMA EXTENSION

DSS adds RDMA extension to existing S3 protocol to meet the bandwidth scalability needs as well as squeezing out every bits/second out of the network to deliver the highest bandwidth to these applications. Although this is a non-standard, this approach allows the ubiquitous S3 protocol which has been adopted by most companies to support high speed data transfers without the overhead of network protocols.

Basically, DSS's concept of using HTTP:S3 w/ RDMA eliminates overheads inherent in NFSv4 protocol by using S3 to fetch whole/entire objects without additional metadata transactions required by NFS and by extending the S3 protocol to support SGL lists, with this approach we can with one transaction, retrieve the data from storage nodes and deliver erasure coded striped data chunks directly to client GPU memory.

In addition, the protocol extension includes the capability to handle erasure code data recovery on the client server system as well as offloading it to the network via Smart NIC or Switch. Using this new approach enables efficient data transfers from device to client GPU memory for best in class storage performance

IV. PERFORMANCE BENCHMARK

First generation DSS presented simple Object Storage and Key Value based NVMe-oF Storage Targets optimized for large data transfers using AMD based Dell R7525 Storage Servers. With Gen2 Architecture, DSS is able to increase performance by almost 58.2% as shown in the table below. Eliminating additional data transfer on the DSS Cluster and moving data via RDMA directly on to host memory, we have significantly improved the throughput of the entire system and thus scale the storage cluster.

We used Elbencho benchmark tool created by BeeGFS for these tests since it is open sourced and easy to understand the scope of benchmark. The CPU Utilization when Elbencho benchmark was ran on DGX-A100 client node was about 8% for Gen2 @39.4GB/s and 19% for Gen1 @20.3GB/s bandwidth achieved. We should note that Gen1 uses TCPIP protocol and incurs much higher CPU utilization versus RDMA on Gen2. This leaves plenty of spare CPU resources for the AI Training or similar application that will need to run on the same DGX-A100 machine to process the incoming data.

In addition, this storage data transfer can be entirely directed to GPU Memory and reduce any system memory contention as well as data copy overheads. For these benchmark tests, we mostly focus our attention to READ Bandwidth performance since that will be the predominant work load for these types of applications (WRITE ONCE followed by READ MULTIPLE). These benchmarks were done with 1Mbyte S3 Objects which were evenly distributed across the cluster nodes.

DSS	Benchmark	# of client nodes	DSS Cluster Nodes	Test Type	Total Files/ Objects	Avg BW (GB/s)	Dell CPU Util	Dell BW (GB/s)	Dell R6525 CPU Util	Dell R6525 BW (GB/s)	DGX A100 CPU Util	DGX A100 BW (GB/s)
Gen1	Elbencho	6	4	GET	60000	100	28%	14.1	8%	15	19%	20.3
Gen2	Elbencho	6	4	GET	60000	158.2	15%	19.7	3%	20.9	8%	39.4

Figure 5: DSS Gen1 vs Gen2 Performance

We utilized 3 different types of client servers to show that CPU Utilization on clients for DSS Gen2 solution significantly reduces CPU overheads on both Intel Xeon and AMD CPU based servers. Notably, on DGX-A100 system only 8% CPU is utilized while receiving/handling 39.4 GB of incoming data from Gen2 storage cluster.

We also ran optimized Lustre storage cluster on the same set of Dell R7525 servers and following results were shown.

System	Benchmark	# of Client Nodes	# of Server Nodes	Test Type	Total Files	Avg BW (GB/s)	CPU Util	BW (GB/s)
Gen2	Elbencho	6	6	READ	1000	237	16	39.5
Lustre	Elbencho	6	6	READ	1000	147	55	24.5

Figure 6: Lustre vs DSS Gen2 6 Node Performance

In this benchmark test, we used 6 Nodes due to additional 2 MDS nodes that are required for Lustre and thus although we were able to achieve 147 GB/s bandwidth which is close to DSS Gen2 numbers if measured on per node basis but we needed 2 additional MDS servers to achieve it. Also, client server CPU utilization for Lustre seems to be much higher than DSS Gen2 which uses the simpler S3 protocol instead.

V. DSS GEN3 PREVIEW

A. *Gen5 HW Platform Support*

Utilizing the next generation servers, PCIe-Gen5 Samsung SSDs and 400 Gb/s NICs, DSS will be able to continue to scale along with latest hardware platforms. We estimate that Gen5 platform may be able to double the current Gen4 platform performance.

In addition, DSS Gen3 will utilize existing CMB on SSDs to transfer data directly from SSDs to Host GPU/Memory. This mechanism will eliminate potential server system DRAM bandwidth congestion and allow full bandwidth transfers from SSDs to GPUs to achieve the highest performance possible.

Furthermore, an intelligent QoS mechanism will allow for sharing of the bandwidth to support multiple concurrent bandwidth intensive applications.

B. *Samsung SmartSSD based Acceleration*

Using intelligent SSD devices such as SmartSSD, DSS Gen3 platform will provide acceleration features to filter and transform stored object data on the devices itself before presenting it to client.

C. *WRITE Path Enhancements*

In order to improve DSS Write Performance path, existing Key Value Storage software system will be redesigned and upgraded. This new Key Value software was architected based on our past research and experience in benchmarking our DSS solution.

VI. CONCLUSION

DSS Gen2/3 architecture continues to improve with each generation. The major objective is to enable latest and greatest hardware platforms and storage devices such as Samsung SSDs to scale and provide the highest bandwidth utilization possible to support major Big Data applications such as AI/ML which can easily consume many petabytes of data for sophisticated training operations and intelligent analytics for businesses, security/surveillance and other use cases. As of today, petabyte scale data sets can take months to process which is a significant barrier for doing any type of development or meaningful research efforts. Although there are multiple technologies that need to come together to shorten the test cycle, storage performance is a major obstacle that must be overcome. DSS technology attempts to reduce the overall software and system complexity by focusing on optimizing READ operations for the highest achievable performance using existing hardware platforms and devices.

REFERENCES

[1] "DSS, The Open Source [P/T]erabyte Scale Storage Engine for High Bandwidth Applications – Samsung – Memory Solutions Lab (samsungmsl.com).

[2] Stephen Pritchard, (2020, April 17), High Performance Object Storage: What's driving it ? High-performance object storage: What's driving it? (computerweekly.com)

[3] "NVM Express," https://nvmexpress.org/.

[4] "Samsung Key Value Document," https://www.samsung.com/ semiconductor/global.semi.static/Samsung Key Value SSD enables High Performance Scaling-0.pdf.

[5] Z. Guz, H. H. Li, A. Shayesteh, and V. Balakrishnan, "NVMe over-fabrics performance characterization and the path to low overhead flash disaggregation," in Proceedings of the 10th ACM SYSTOR, 2017.

[6] "NKV Library," https://github.com/OpenMPDK/NKV.

[7] "MinIO object storage cluster," https://min.io/resources/docs/ MinIO-high-performance-object-storage.pdf.

[8] "Amazon S3," https://docs.aws.amazon.com/AmazonS3/latest/ dev/s3-dg.pdf.

[9] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. Long, and C. Maltzahn, "Ceph: A scalable, high-performance distributed file system," in Proceedings of the 7th OSDI SYMPOSIA, 2006.

[10] H. Lim, D. Han, D. G. Andersen, and M. Kaminsky, "MICA: A holistic approach to fast in-memory key-value storage," in 11th NSDI, 2014.

[11] X. Wu, Y. Xu, Z. Shao, and S. Jiang, "LSM-Trie: An LSM tree-based ultra-large key-value store for small data items," in USENIX ATC, 2015.

[12] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's highly available key-value store," in ACM SIGOPS, 2007.

[13] Y. Jin, H.-W. Tseng, Y. Papakonstantinou, and S. Swanson, "KAML: A flexible, high-performance key-value ssd," in IEEE International Symposium on HPCA, 2017.

[14] Y. Kang, R. Pitchumani, P. Mishra, Y.-s. Kee, F. Londono, S. Oh, J. Lee, and D. D. Lee, "Towards building a high performance, scale-in key-value storage system," in Proceedings of the 12th ACM SYSTOR, 2019.

[15] T. Bisson, K. Chen, C. Choi, V. Balakrishnan, and Y.-s. Kee, "Crail-KV: A high-performance distributed key value store leveraging native KV-ssds over NVMe-oF," in IEEE 37th IPCCC, 2018.