High-Performance Multi-Rail Erasure Coding Library over Modern Data Center Architectures: Early Experiences *

Haiyang Shi, Xiaoyi Lu, Dipti Shankar, Dhabaleswar K. (DK) Panda

The Ohio State University, {shi.876, lu.932, shankar.50, panda.2}@osu.edu

Abstract

Various hardware-based Erasure Coding (EC) schemes have been proposed [5, 6, 8, 12–14] to leverage the advanced compute capabilities on modern data centers. Currently, there is no unified and easy way for distributed storage systems to fully exploit multiple devices such as CPUs, GPUs, and network devices (i.e., multi-rail support) to perform EC operations in parallel. In this paper, we validate that it is time to design an unified library to efficiently exploit heterogeneous EC coders. HDFS co-designed with our proposed library outperforms the write performance of replication scheme and the default HDFS EC coder by 2.7x - 6.1x and 2.4x - 3.3x, respectively, and improves the performance of read with failure recoveries by up to 2.6x and 5.1x compared to the replication scheme and the default HDFS EC coder, respectively.

1 MOTIVATION

Prevalent EC techniques such as Reed-Solomon (RS) [11, 15] codes and its variations have been widely used in many distributed storage systems [1–3, 9, 10, 16]. To overcome the high computational costs involved with RS erasure coding, two broad categories of coders have been proposed to take advantage of modern hardware capabilities: (1) *EC-Onload*, where optimized host-based libraries such as Jerasure [14] and Intel ISA-L [5] are employed, and, (2) *EC-Offload*, wherein the EC computation tasks can be offloaded to accelerators (e.g., GPGPU [4]) or high-performance network devices (e.g., Host Channel Adapters (HCA) of Mellanox ConnectX-4 and later [7]). Although these hardware-optimized libraries can potentially facilitate EC to be employed as a viable choice for fault-tolerance in modern distributed storage systems, none

of current-generation storage systems can exploit multiple devices like CPUs, GPUs, and network devices (i.e., *Multi-Rail Support*) in parallel on modern data centers. These drawbacks leave the available high-speed hardware significantly under-utilized and adversely affect the performance of the EC-based storage system.

2 OUR APPROACH

To provide unified and high-speed EC APIs by exploiting heterogeneous hardware-optimized EC coders, our proposed design comprises on three fundamental building blocks: (1) The multi-rail based EC scheme enables upper layer applications to exploit the heterogeneous EC capabilities of modern data centers in parallel, (2) The uniform and asynchronous interfaces abstract all the functionalities required by the distributed storage system to access different types of hardware-specific coders, and can be used to design a heterogeneity-aware I/O pipeline that can match the high-throughput requirements of distributed storage systems, and (3) We co-design HDFS 3.x with proposed library to demonstrate the effectiveness and productivity of exploiting it in distributed storage systems.

3 PRELIMINARY RESULTS

We evaluate our proposed library with HDFS on 16 nodes. As shown in Figure 1, the TestDFSIO benchmark illustrates that the co-designed I/O pipelines with proposed library can outperform default HDFS with both replication and EC schemes. Compared to the 4-way replication and the default EC scheme, the co-designed HDFS can improve the write performance by 3.7x - 6.1x and 2.4x - 3.3x, respectively, and can improve the performance of read with failure recoveries by up to 2.6x and 5.1x, respectively.

*This research is supported in part by National Science Foundation grants #CNS-1419123, #IIS-1447804, #ACI-1450440, #CNS-1513120, and #IIS-1636846.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SoCC '18, October 11–13, 2018, Carlsbad, CA, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6011-1/18/10.

<https://doi.org/10.1145/3267809.3275472>

Figure 1: Performance Comparison on 16 nodes.

REFERENCES

- [1] Apache Hadoop 3.0.0-alpha2 2017. Apache Hadoop 3.0.0-alpha2. <http://hadoop.apache.org/docs/r3.0.0-alpha2/>.
- [2] Backblaze Reed-Solomon 2015. Backblaze Reed-Solomon. <https://www.backblaze.com/open-source-reed-solomon.html>.
- [3] Ceph Erasure Coding 2016. Ceph Erasure Coding. <http://docs.ceph.com/docs/master/rados/operations/erasure-code/>.
- [4] Matthew Curry, Anthony Skjellum, H Lee Ward, and Ron Brightwell. 2011. Gibraltar: A Reed-Solomon Coding Library for Storage Applications on Programmable Graphics Processors. In *Concurrency and Computation: Practice and Experience*, Vol. 23. 2477–2495.
- [5] Intel Intelligent Storage Acceleration Library (Intel ISA-L) 2016. Intel Intelligent Storage Acceleration Library (Intel ISA-L). <https://software.intel.com/en-us/storage/ISA-L>.
- [6] Aleksei Marov and Andrey Fedorov. 2016. Optimization of RAID Erasure Coding Algorithms for Intel Xeon Phi. In *Networking, Architecture and Storage (NAS), 2016 IEEE International Conference on*. IEEE, 1–4.
- [7] Mellanox. 2016. HDFS Erasure Coding Offload Plugin. <https://github.com/Mellanox/EC/tree/master/HDFS>.
- [8] Mellanox. 2016. Understanding Erasure Coding Offload. <https://community.mellanox.com/docs/DOC-2414>.
- [9] Subramanian Muralidhar, Wyatt Lloyd, Sabyasachi Roy, Cory Hill, Ernest Lin, Weiwen Liu, Satadru Pan, Shiva Shankar, Viswanath Sivakumar, Linpeng Tang, et al. 2014. f4: Facebook’s Warm BLOB Storage System. In *Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation*. USENIX Association, 383–398.
- [10] Michael Ovsianikov, Silvius Rus, Damian Reeves, Paul Sutter, Sriram Rao, and Jim Kelly. 2013. The Quantcast File System. *Proceedings of the VLDB Endowment* 11 (2013), 1092–1101.
- [11] James S Plank. 2005. Optimizing Cauchy Reed-Solomon Codes for Fault-tolerant Storage Applications. *University of Tennessee, Tech. Rep. CS-05-569* (2005).
- [12] James S. Plank, Kevin M. Greenan, and Ethan L. Miller. 2013. Screaming Fast Galois Field Arithmetic Using Intel SIMD Instructions. In *11th USENIX Conference on File and Storage Technologies (FAST 13)*. USENIX Association, San Jose, CA, 298–306.
- [13] James S Plank, Jianqiang Luo, Catherine D Schuman, Lihao Xu, Zooko Wilcox-O’Hearn, et al. 2009. A Performance Evaluation and Examination of Open-Source Erasure Coding Libraries for Storage. In *Proceedings of the 7th Conference on File and Storage Technologies (FAST ’09)*. USENIX Association, Berkeley, CA, USA, 253–265. <http://dl.acm.org/citation.cfm?id=1525908.1525927>
- [14] James S Plank, Scott Simmerman, and Catherine D Schuman. 2008. Jerasure: A Library in C/C++ Facilitating Erasure Coding for Storage Applications. (2008).
- [15] Irving S Reed and Gustave Solomon. 1960. Polynomial Codes Over Certain Finite Fields. *J. Soc. Indust. Appl. Math.* 8, 2 (1960), 300–304.
- [16] Dipti Shankar, Xiaoyi Lu, and Dhabaleswar K Panda. 2017. High-Performance and Resilient Key-Value Store with On-line Erasure Coding for Big Data Workloads. In *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 527–537.