Designing High-Performance, Resilient and Heterogeneity-Aware Key-Value Storage for Modern HPC Clusters [*]

Dipti Shankar
Advisor: Dr. Dhabaleswar K. Panda, Co-Advisor: Dr. Xiaoyi Lu
Department of Computer Science and Engineering, The Ohio State University
{shankar.50, panda.2, lu.932}@osu.edu

ABSTRACT

Distributed key-value stores are being increasingly used to accelerate Big Data workloads on modern HPC clusters. The advances in HPC technologies (e.g., RDMA, SSDs) has directed several efforts towards employing hybrid storage with RDMA, for designing high-performance key-value stores. With this as basis, in my research, I take a holistic approach to designing a high-performance key-value storage system for HPC clusters that can maximize end-to-end performance while ensuring data resilience, that encompasses: (1) RDMA-enabled networking, (2) high-speed NVMs, and, (3) heterogeneous compute capabilities, available on current HPC systems. Towards this, I introduce RDMA-aware designs to enable: (1) non-blocking API semantics for designing high-performance client-side read/write pipelines, (2) fast online erasure coding for memory-efficient resilience, and, (3) SIMD-aware server-side accelerations; to enable Big Data applications to optimally leverage hybrid key-value stores in HPC environments.

1 INTRODUCTION

With the recent emergence of in-memory computing into mainstream Big Data analytics, distributed key-value stores have become vital for accelerating various data processing workloads, including, online analytical workloads and offline data-intensive workloads. For instance, distributed and scalable key-value stores like Memcached [5] are often used to cache popular data items, to improve overall performance while reducing the load on the backend database systems. On the other hand, for offline data-intensive workloads, distributed key-value stores are being increasingly used to design efficient I/O staging and caching solutions, in order to alleviate the parallel filesystem bottleneck on modern HPC clusters [13]. Several research works have focused on exploiting Remote-Direct-Memory-Access (RDMA) feature available on network interconnects, such as InfiniBand [1], to improve the response time of in-memory key-value stores [6, 7] on HPC clusters; while 'RAM+SSD' hybrid storage has been explored to increase their data retention [9]. With the recent advancements in high-performance computing (HPC) technologies, we believe that there is scope for further improving the performance of such high-performance and hybrid key-value stores.

2 PROBLEM STATEMENT

Existing high-performance and hybrid key-value stores for HPC clusters are constrained by the blocking store/retrieve semantics. Thus, they may not be able to exploit the available advanced network, storage and compute capabilities to the fullest. This complexity increases further if we introduce high data availability and resilience requirements into the design of these key-value stores. Thus, to deliver optimal performance, we need to consider designs that can focus beyond optimizing the communication engine in key-value stores for better latencies. Specifically, it is necessary to consider how we can enable data-intensive key-value store workloads to define more efficient data movement pipelines, so as to have fast access to remote distributed data. Towards this end, it is therefore vital to address the following broad challenges:

1. Can we design key-value store APIs that can truly leverage the one-sided semantics of RDMA, while conforming its data movement semantics to those in general in-memory and hybrid key-value stores?
2. Can we ensure resilience and data availability for these key-value stores, while enabling high-performance and memory efficiency?
3. Can we leverage the heterogeneous compute capabilities (e.g., GPU) and emerging persistent memory technologies (i.e., NVRAM) to accelerate key-value store servers along with maximizing end-to-end performance?
4. If we can design an RDMA-aware and hybrid key-value storage system with the above features, can it benefit Online and Offline Big Data analytical workloads?

3 RESEARCH HIGHLIGHTS

Along the direction discussed in the problem statement (Section 2), this research is focused on taking a holistic approach that can fully exploit resources on modern HPC systems to cater to both performance and resilience needs of online and offline key-value store workloads. Figure 1 depicts the research framework that is employed to address the challenges highlighted above.

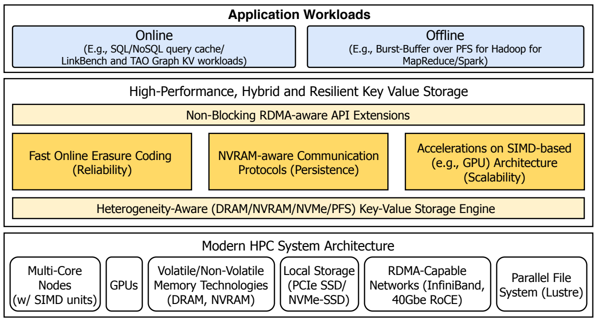
Application Workloads
Online
(E.g., SQL/NoSQL query cache/ LinkBench and TAO Graph KV workloads)
Offline
(E.g., Burst-Buffer over PFS for Hadoop for MapReduce/Spark)
High-Performance, Hybrid and Resilient Key Value Storage
Non-Blocking RDMA-aware API Extensions
Fast Online Erasure Coding (Reliability)
NVRAM-aware Communication Protocols (Persistence)
Accelerations on SIMD-based (e.g., GPU) Architecture (Scalability)
Heterogeneity-Aware (DRAM/NVRAM/NVMe/PFS) Key-Value Storage Engine
Modern HPC System Architecture
Multi-Core Nodes (w/ SIMD units)
GPUs
Volatile/Non-Volatile Memory Technologies (DRAM, NVRAM)
Local Storage (PCIe SSD/ NVMe-SSD)
RDMA-Capable Networks (InfiniBand, 40Gbe RoCE)
Parallel File System (Lustre)

Figure 1: Research Framework

[*] This research is supported in part by National Science Foundation grants #CNS-1513120, #IIS-1636846, and #CCF-1822987.

For all of our designs so far, we have performed extensive evaluations on different small-scale clusters: (1) 8-core Intel Westmere, InfiniBand QDR, 24 GB RAM, and PCIe-SSDs (144 nodes / in-house), (b) 28-core lntel Haswell, InfiniBand EDR, 256 GB DRAM, with NVMe-SSDs (20 nodes / in-house); along with production-scale clusters, such as, SDSC Gordon and SDSC Comet [2] (all equipped with Lustre PFS).

3.1 Non-Blocking RDMA-aware API Semantics

Towards addressing the first challenge, in [10], we introduce non-blocking API extensions to the RDMA-Memcached client, that allows the user to separate the request issue and completion phases. This facilitates overlapping opportunities via one-sided characteristics of the underlying RDMA communication engine, while conforming to the basic Set/Get semantics. To demonstrate its applicability, we study the performance of employing these non-blocking APIs to hide the SSD I/O overhead in the SSD-assisted RDMA-Memcached [9], and show that we can achieve near in-memory client-side latencies by overlapping request/response phases across multiple concurrent Set/Get operations.

3.2 Fast Online Erasure Coding with RDMA

Towards addressing the second challenge, in [12], we model and analyze the overhead of employing memory-efficient resilience techniques, such as Reed-Solomon erasure codes (RS-EC), in an online (i.e, 'on-the-fly') fashion. Based on this, we propose an RDMA-aware design that can leverage our non-blocking API semantics to enable the overlap of the encoding/decoding compute phases with the scatter/gather communication pattern of the data/parity chunks per-key-value-item. We analyze the possible RDMA-aware online EC designs by enabling the offloading of the encode/decode phases at both the client and the server. We study the performance of these approaches in-depth using the YCSB benchmark [3] that mimics online key-value store workloads.

3.3 Co-Designing Key-Value Store-based Burst Buffer over Parallel Filesystems

Towards addressing the fourth challenge (in-part), we attempt to perform a case study to illustrate the advantages of leveraging our proposed designs for offline Big Data analytical workloads on HPC clusters. We design, Boldio [11], a hybrid and resilient key-value store-based burst-buffer system over Lustre. We demonstrate how significant benefits can be achieved for Hadoop I/O workloads relying on PFS; and also contrast the performance of asynchronous RDMA-based replication and RDMA-aware online EC for resilience.

3.4 Ongoing Research Works

Towards addressing the third challenge, as a part of this thesis, we are exploring opportunities for employing emerging persistent memories (NVRAM), such as PCM, 3DXpoint, etc. Their byte-addressability opens up opportunities for leveraging RDMA to access and persistent data in remote memory [4]; enabling us to consider incorporating 'RDMA+NVRAM' capabilities into key-value stores via high-performance remote persistence and access protocols. On the other hand, SIMD capabilities of HPC compute resources (e.g, GPU) have been explored to significantly improve key-value server throughput [14], but are not inherently optimized to benefit client-side performance. Based on this, we plan on exploring end-to-end SIMD-aware designs.

3.5 Software Artifacts

The SSD-assisted and RDMA-enhanced Libmemcached/Memcached designs, along with proposed non-blocking APIs, are available as a part RDMA for Memcached software package, under NowLab's High-Performance Big Data (HiBD) project [8]. We are progressively working on integrating the online EC designs into this.

4 CONCLUSION

In my research, I address several challenges in designing and building high-performance and resilient key-value storage systems for modern HPC clusters. Towards this, a holistic approach is employed to exploit the HPC technological advances, including, network (i.e., RDMA), storage (SSD/NVRAM), and compute (multi-core CPUs, GPUs). Based on this, to maximize the end-to-end performance of hybrid key-value stores, RDMA-aware designs for: (1) non-blocking key-value store API semantics, (2) fast memory-efficient resilience, were introduced; with SIMD- and NVRAM-aware optimizations in-the-works. In addition, we demonstrate the potential performance gains of the proposed designs with online (e.g, YCSB) and offline (e.g, key-value store-based burst-buffer over Lustre for Hadoop I/O) workloads on small-scale and production-scale HPC clusters.

REFERENCES

[1] 2018. Infiniband Trade Association. http://www.infinibandta.org/

[2] San Diego Supercomputer Center. 2018. SDSC: HPC Systems. http://www.sdsc.edu/services/hpc/hpc_systems.html.

[3] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears. 2010. Benchmarking Cloud Serving Systems with YCSB. In *The Proceedings of the ACM Symposium on Cloud Computing (SoCC '10)*. Indianapolis, Indiana.

[4] Intel Corporation. 2018. pmdk: Persistent Memory Development Kit. https://github.com/pmem/pmdk/.

[5] Brad Fitzpatrick. 2004. Distributed Caching with Memcached. *Linux Journal* 2004 (August 2004), 5–. Issue 124.

[6] Jithin Jose, Hari Subramoni, Miao Luo, Minjia Zhang, Jian Huang, Md. Wasi-ur Rahman, Nusrat S. Islam, Xiangyong Ouyang, Hao Wang, Sayantan Sur, and Dhabaleswar K. Panda. 2011. Memcached Design on High Performance RDMA Capable Interconnects. In *Proceedings of the 2011 International Conference on Parallel Processing (ICPP '11)*. Washington, DC, USA.

[7] A. Kalia, M. Kaminsky, and D. G. Andersen. 2014. Using RDMA Efficiently for Key-Value Services. In *Proceeding of SIGCOMM '14*.

[8] OSU NowLab. 2018. High-Performance Big Data (HiBD). http://hibd.cse.ohio-state.edu.

[9] Xiangyong Ouyang, N.S. Islam, R. Rajachandrasekar, J. Jose, Miao Luo, Hao Wang, and D.K. Panda. 2012. SSD-Assisted Hybrid Memory to Accelerate Memcached over High Performance Networks. In *Proceeding of the 41st International Conference on Parallel Processing (ICPP '12)*. 470–479.

[10] D. Shankar, X. Lu, N. Islam, M. Wasi-Ur-Rahman, and D. K. Panda. 2016. High-Performance Hybrid Key-Value Store on Modern Clusters with RDMA Interconnects and SSDs: Non-blocking Extensions, Designs, and Benefits. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 393–402.

[11] D. Shankar, X. Lu, and D. K. Panda. 2016. Boldio: A Hybrid and Resilient Burst-Buffer Over Lustre for Accelerating Big Data I/O. In *2016 IEEE International Conference on Big Data (Big Data)*. 404–409.

[12] D. Shankar, X. Lu, and D. K. Panda. 2017. High-Performance and Resilient Key-Value Store with Online Erasure Coding for Big Data Workloads. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 527–537.

[13] Teng Wang, S. Oral, Yandong Wang, B. Settlemyer, S. Atchley, and Weikuan Yu. 2014. BurstMem: A High-Performance Burst Buffer System for Scientific Applications. In *2014 IEEE International Conference on Big Data*.

[14] Kai Zhang, Kaibo Wang, Yuan Yuan, Lei Guo, Rubao Lee, and Xiaodong Zhang. 2015. Mega-KV: A Case for GPUs to Maximize the Throughput of In-Memory Key-Value Stores. *Proceedings of the VLDB Endowment* 8, 11 (2015), 1226–1237.