ReFlex4ARM: Supporting 100GbE Flash Storage Disaggregation on ARM SoC*

Minghao Xie
Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, United States
mhxie@ucsc.edu

Chen Qian
Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, United States
cqian12@ucsc.edu

Heiner Litz
Computer Science and Engineering
University of California, Santa Cruz
Santa Cruz, United States
hlitz@ucsc.edu

***Abstract*—Flash Disaggregation enables to share flash storage across the data center, improving resource utilization and reducing the total cost of ownership (TCO). Previous work on flash disaggregation utilized costly server processors leaving significant headroom for optimizing TCO. In this work, we develop a new flash disaggregation system based on a cost-effective and power-efficient ARM-based Smart NIC. This work introduces our architecture and provides a comprehensive evaluation outperforming previous work in TCO by 2.57x.**

I. Introduction

In recent years, flash has been established as an important storage tier in data centers due to its high performance. The rising IOPS demand of applications and the strict latency requirements of users facing cloud workloads render flash the prime choice for storing data. However, equipping every server with an individual flash device is costly and reduces efficiency due to imbalanced resource requirements of applications. Flash disaggregation addresses this problem by sharing all flash resources within the data center among all tenants and by allowing to scale flash and compute resources independently. However, to enable disaggregation, remote storage needs to be accessible with high performance at low cost. Legacy remote storage access protocols such as iSCSI are incapable of handling the millions of IOPS that modern SSDs can provide and furthermore introduce a high latency penalty. ReFlex [1] removes the performance overhead by tightly integrating the networking layer and storage layer by employing IX data plane architecture with the high-performance storage access framework SPDK. It makes a single mid-level Xeon Core be able to handle up to 850K IOPS in a 1024-byte random read test. Nevertheless, the cost of flash storage nodes based on Xeon machines combined with their high energy consumption is in non-optimal in terms of TCO.

We developed *ReFlex4ARM*, a flash disaggregation system combining an 100GbE NIC with low cost ARM processors and an NVMe storage interface to provide remote access to flash at 2M IOPS with sub $100\mu s$ access latency at low cost. We utilize the Broadcom Stringray PS1100R [2] supporting eight ARM A72 cores integrated with a NIC, memory controller and PCIe interfaces on a single chip. By configuring it as a PCI-e root complex connecting to up to four SSDs, we can offload all storage disaggregation tasks from the host, reducing capital expenditure as well as improving power consumption.

II. Challenges

While SmartNICs such as Broadcom's PS1100R improve power consumption and TCO, their ARM cores provide significantly less compute performance and memory bandwidth than x86 based server processors. Developing a flash disaggregation based on an ARM SoC, therefore, requires a highly optimized and efficient design while not compromising on portability to alleviate adoption. In the following we describe three challenges we had to overcome to develop our ARM based solution.

Portability: Our design is based on ReFlex [1] which is limited to run on x86_64 machines. ReFlex depends on x86 specific inline assembly code for performance critical sections. To enable the ARM ISA we have refactored ReFlex replacing all architecture specific code with generic implementation. In the cases where inline assembly is required we modify the code to emit x86 or ARM specific assembly specified at compile time enable portability across hardware. To enable fast adoption, ReFlex has been updated to support Ubuntu 19.04 and recent versions of DPDK (v19.05) and SPDK (v19.04).

Performance: Achieving x86 like performance on ARM processors is challenging as the ARM A72's micro-architecture does not provide the same level or performance than modern x86 architectures. To increase the throughput of ReFlex4ARM, we resort to domain specific hardware acceleration. In particular, we leverage custom hardware for CRC computation, flow direction, checksum offloading and direct memory access. The combination of ARM specific code optimisations and hardware acceleration capabilities enables ReFlex4ARM to deliver high IOPS at low latency. As shown in Figure 1, we saturated our four-SSD system by only using four cores. Utilizing all eight cores improves performance for 4K requests, reaching the maximum IOPS rate of our four SSDs. Deploying faster SSDs we expect ReFlex4ARM to provide up to 3M remote IOPS. In addition to throughput, datacenter operators are generally also interested in latency especially at the tail [3]. In Figure 2 we show the latency distribution of a

This work was supported in part by NSF grant number IIP-1266400 and IIP-1841545 and by the industrial partners of the Center for Research in Storage Systems.

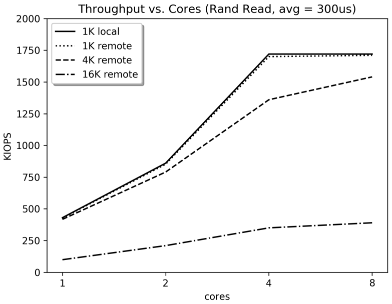

Fig. 1. Throughput Scaling

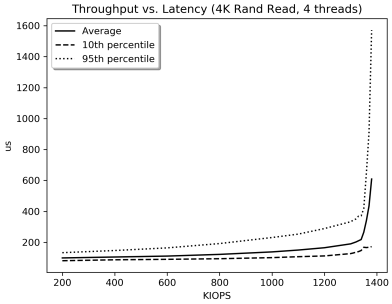

Fig. 2. 4k Random Read Latency Distribution

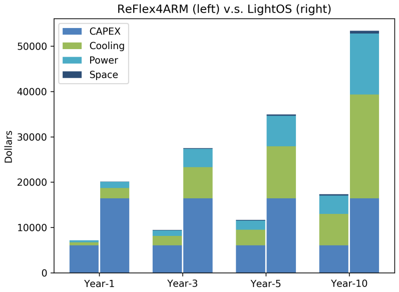

Fig. 3. TCO Analysis

4k random read workload utilizing four cores. ReFlex4ARM can enforce a 95th tail latency of less than 300 μs at 420K IOPS and 400 μs at 1.4 million IOPS.

TCO: TCO represents the primary optimization metric for data center operators [4]. We have developed a cost model for ReFlex4ARM and compare it with Lightbits [5], a separate storage disaggregation platform that offers 5 MIOPS at 4K requests. We utilize the same cost model to calculate the TCO for Lightbits and ReFlex4ARM finding that ReFlex reduces TCO by 2.6×. As shown in Figure 3, ReFlex4AM provides better TCO including both CAPEX and OPEX at 1.32×10^{-6} dollar per million 4K IOs. When comparing Reflex on ARM vs. Intel Xeon, we provide an increase of 1.57× IO operations for the same cost.

Table 1: Hardware Configuration Overview

Solution	Front-end	Memory	SSDs	NIC
ReFlex4ARM	Stingray	16GB	16x	200Gbps
LightOS	Xeon	128GB	24x	200Gbps

III. Discussion

ReFlex4ARM follows the original ReFlex design, leveraging key techniques such as system call batching, process to completion and zero copy DMA transfers between network and storage hardware. ReFlex4ARM improves over the original design by addressing various performance issues, by adding ARM support and by extending the design to support multiple SSDs and modern 100GbE networking hardware. Our source code has been published at https://github.com/mhxie/reflex4arm.

Storage Scalability: To improve the throughput of our flash disaggregation system we developed support for 100GbE NICs and also introduced multi SSDs support. The PS1100R supports 16 lanes of PCIe 3.0 which can drive four 4-lane SSDs. ReFlex4ARM supports multiple SSDs by sharding storage capacity across devices. The control plane is responsible to assign a port number for each SSD and negotiate with client to make connections up to the number of SSDs. The leading bits of requested logical block address are used to determine the corresponding SSD namespace.

Network Performance Tuning: Theoretically, an 100GbE NIC transfers up to 3.2Mpps utilizing 4K packets. To support these high packet rates we performed comprehensive networking parameter tuning to minimize overheads. In particular, we enabled Jumbo frames of 6000 bytes to optimize 4K-performance, increased the per-connection TCP sending buffer and TCP window size with large scale options. Besides, we also increased the amount of adaptive batching and compacted request structure to improve the memory allocation to further increase performance.

Flow SLO Configuration: To extend the use of ReFlex we defined a new configuration interface that can be utilized to remotely specify the service level objective (SLO) of a flow. Each flow can be assigned with throughput and tail latency requirements that the ReFlex4ARM scheduler will then enforce at runtime. Our flow SLO interface can be leveraged by the data center control plane to direct flows to particular storage nodes and flexibly configure their SLO requirements at runtime.

IV. Future Work

ReFlex4ARM provides an efficient dataplane for remote access to Flash. To deploy ReFlex in a datacenter cluster, ReFlex4ARM needs to be combined with a control plane to manage Flash resources across machines and optimize the allocation of Flash IOPS and capacity. We are implementing a control plane with a high performance job-level scheduler as part of future work.

V. Conclusion

Reflex4ARM is a TCO-optimized cross-platform flash disaggregation solution. It enables new storage applications that requires high-performance networked storage at low cost enforcing service level objectives.

References

[1] Klimovic, A., Litz, H., & Kozyrakis, C. (2017). Reflex: Remote flash ≈ local flash. ACM SIGPLAN Notices, 52(4), 345-359.

[2] Broadcom Inc., https://www.broadcom.com/products/storage/ethernet-storage-adapters-ics/ps1100r, 2020

[3] Dean, J., & Barroso, L. A. (2013). The tail at scale. Communications of the ACM, 56(2), 74-80.

[4] Barroso, L. A., Hölzle, U., & Ranganathan, P. (2018). The datacenter as a computer: Designing warehouse-scale machines. Synthesis Lectures on Computer Architecture, 13(3), i-189.

[5] lightbits Inc., https://www.lightbitslabs.com/dell-poweredge/, 2020