Case Study: Integration of SALSA, a unified storage software stack developed by IBM Research, with Radian's Zoned Flash SSD

This case study provides an industry-first overview of the integration between a purpose-built storage software stack and a zoned Flash SSD, including block level benchmark results and system level, application benchmark results compared to the same software stack using a near identical FTL-based SSD.

Rapid Integration + Exceptional Performance

Block level benchmarks (fio):
Over 3x improvement in throughput,
50x improvement in tail latencies, and
an est. 3x improvement in Flash wear out
...all at the same time

System level benchmarks (MySQL®/SysBench):
Over 65% improvement in tps,
Over 22x improvement in tail latencies, and
an est. 3x improvement in Flash wear out
...all at the same time

While zoned Flash SSDs can provide significant improvements in system performance, integrating them with a host storage management layer often requires extensive software modifications and development efforts. This document provides an overview of how Radian's Zoned Flash SSD, combined with the flexible architecture of SoftwAre Log Structured Array (SALSA), developed by scientists at IBM Research enabled achieving dramatic system performance improvements with minimal development efforts.

Integration Time Estimate		
	Functional I/O	Performance Tuning
Calendar Time	1 Week	4 Weeks
Person Weeks	2 Weeks	8 Weeks

Integration time estimates started after platform settings and correct software patches were configured

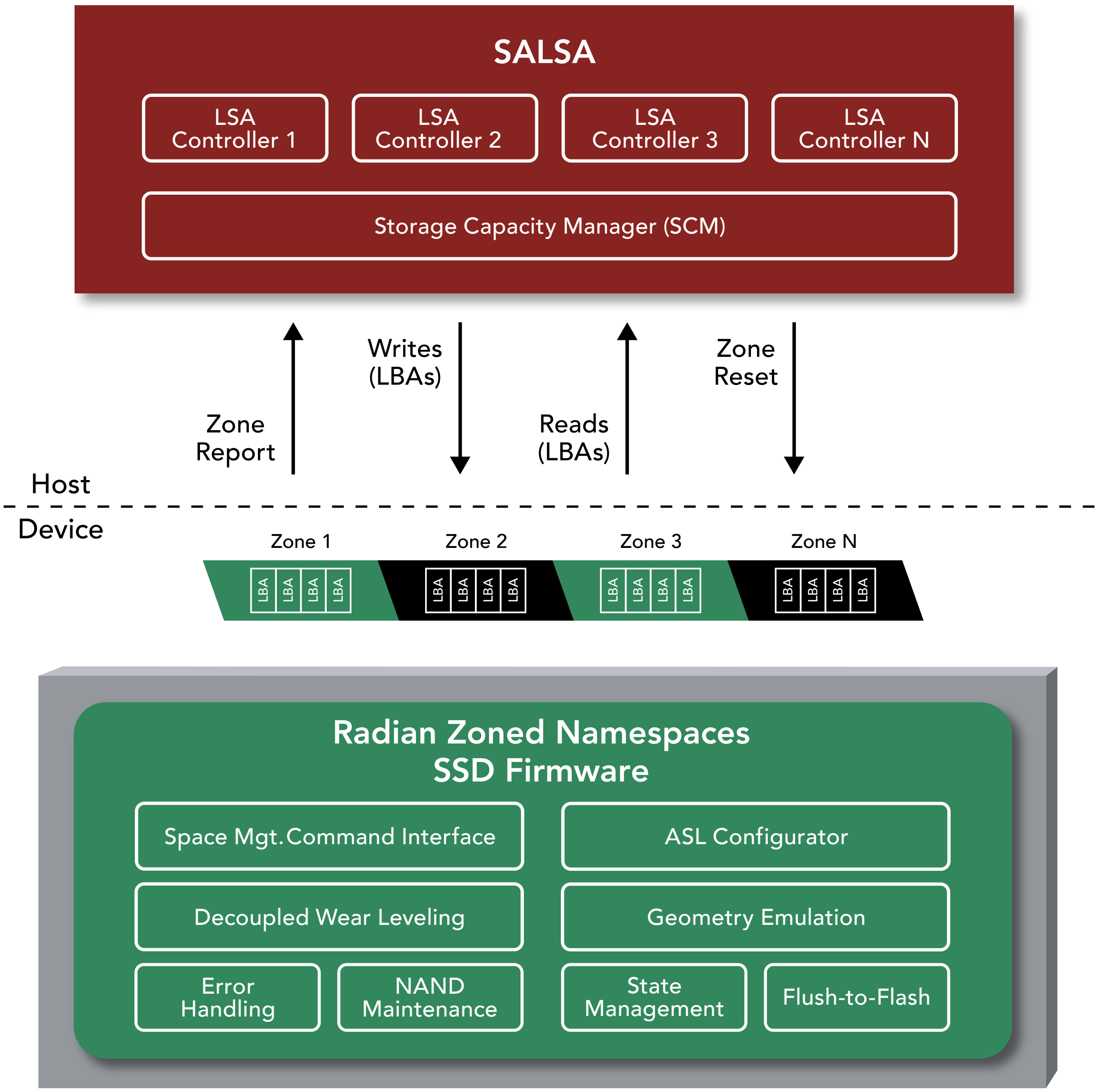
SALSA

SALSA is a versatile, host resident translation layer that can virtualize multiple storage devices (SSDs or HDDs) and be configured to adapt to specific workloads. Targeting the backend of Software-Defined Storage (SDS) frameworks, SALSA can expose a Linux block device that can either be used directly by unmodified applications, or mounted by a traditional Linux filesystem.

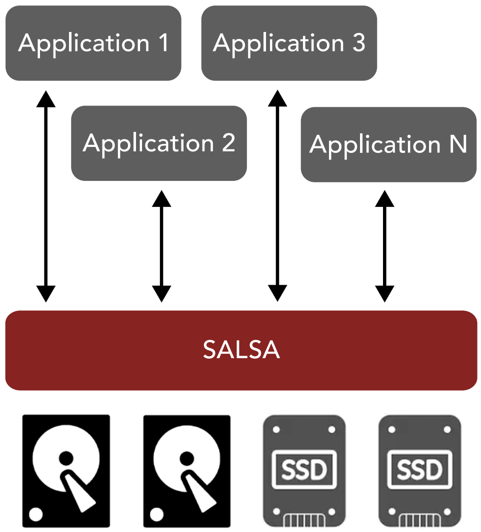
Radian Zoned Flash SSD

The RMS-350 is a U.2 NVMe SSD based upon Radian Memory Systems' implementation of Zoned Namespaces (ZNS) that provides idealized, configurable Flash SSD zones. Routine Flash management processes are cooperative between the device and the host to provide superior system level performance and efficiencies than can be achieved with FTL-based SSDs.

SALSA controls data placement while the Radian Zoned Flash SSD abstracts lower level media management, including geometry and vendor-specific NAND attributes. SALSA controls garbage collection, including selecting which zones to reclaim and where to relocate valid data, while the zoned Flash SSD deterministically performs other NAND management processes.

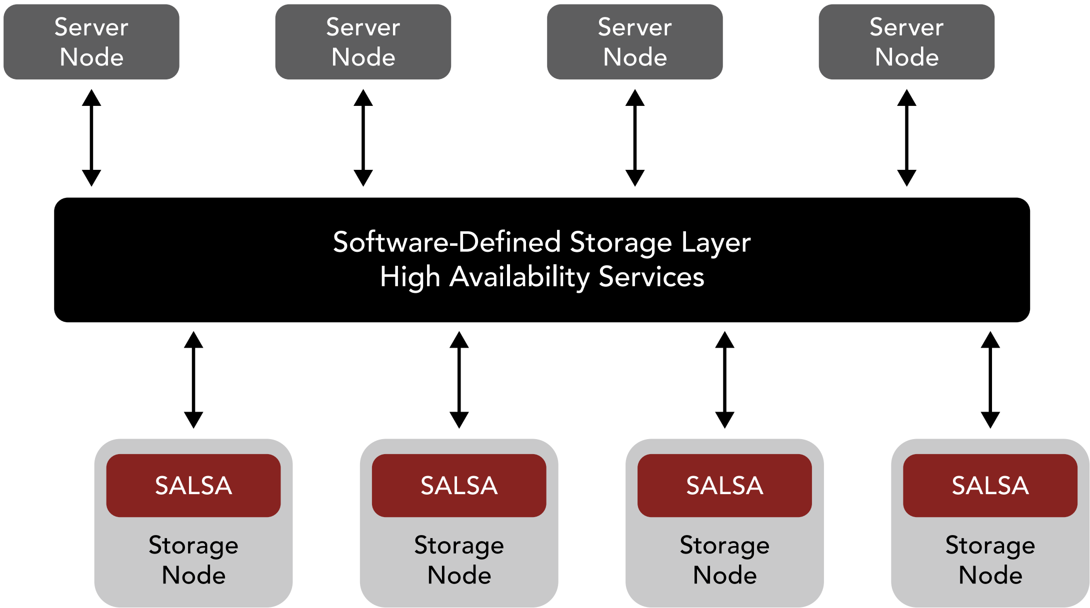


SALSA supports multiple storage device types (different SSDs and HDDs), and can combine them through its translation layer to build hybrid systems.



As Direct-Attached Storage, SALSA can be mounted directly by local file systems or databases to provide performant, cost effective local storage. Its flexible architecture enables concurrently supporting different applications while also optimizing for different workload-specific policies.

As a backend to SDS layers, SALSA can provide disaggregated storage that enables independent scaling of compute and storage. While the distributed SDS layer assumes responsibilities for higher level availability and storage services (data protection, replication, backup, disaster recovery), the SALSA translation layer can locally manage individual storage devices for maximum efficiency.



Rapid Integration

Radian Zoned Flash SSD	SALSA
Idealized Flash	*No changes required to address translation, failure modes, consistency models or core logic*
	Did not need to handle variable zone sizes or zone excursions
	No need to implement Zone Append because Relaxed Write Pointers were provided
	Did not need to account for low level NAND attributes or vendor-specific anomalies
	Did not need to perform NAND error handling or manage bad block replacement
ASL Configurator	Configured drive without writing new mapping code
	Configured isolated regions, zones (segments) and write stripes to best match SALSA architecture and application requirements
	Did not need to account for addressing gaps inside zones or between zones, as zones were presented with contiguous LBAs
Decoupled Wear Leveling & NAND Maintenance	*Did not have to implement wear leveling into cleaning policy or track wear metrics*
	Did not have to assimilate NAND data retention requirements or implement scrubbing processes to prevent unrecoverable errors

Exceptional Performance

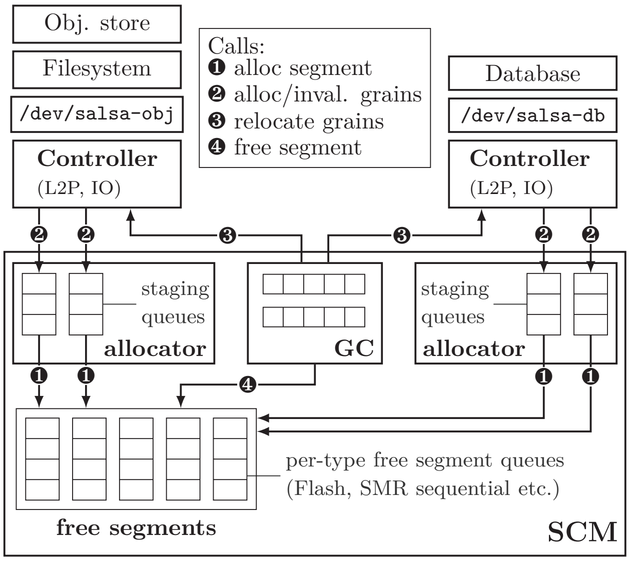
Radian Zoned Flash SSD	SALSA
Idealized Flash	*No need to limit queue depth (no strict write pointer restriction)*
	Zone Append not required so no associated latency penalty creating stripes or from mapping updates
	SALSA did not need to manage low level NAND attributes – doing this over a system bus is inefficient and introduces overhead
ASL Configurator	*Configurable zones minimized write amp., improving throughput and latency*
	Configurable write stripes improved parallelism and throughput
	Configurable iso-boxes minimized collisions to improve tail latencies
Decoupled Wear Leveling & NAND Maintenance	*Did not have to repetitively poll device to query wear leveling and data retention metrics*

SALSA (SoftwAre Log Structured Array)

Developed by IBM Research, SALSA is a host resident storage software layer that can pool multiple storage devices, optimizing functions such as load balancing, while exposing block devices that can be used transparently by applications, e.g., by running a database directly on the device, or by creating a filesystem and running an application on top.

SALSA has been built as a framework that supports multiple types of storage, different policies and algorithms, and a wide range of configuration options. Based upon a log-structured architecture that separates space management from storage policy, SALSA can accommodate different applications, each with its own policy, using the same storage pool without requiring any application modifications.

There are two main components in SALSA: the storage capacity manager (SCM), which is responsible for managing the underlying storage, and one or more controllers that operate on the SCM. The SCM is a common substrate that implements storage provisioning to controllers, garbage collection (GC), and other common functions. The controllers are responsible for implementing the storage policy, performing I/O, and mapping the logical (application) space to the physical space.

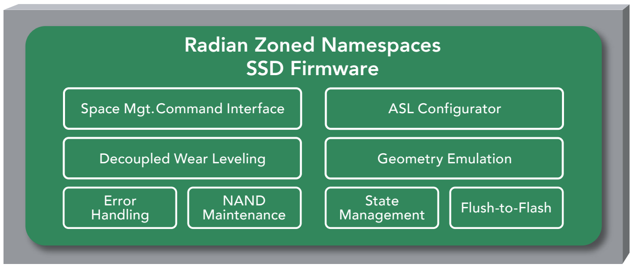


By residing on the host instead of the device, SALSA enables a global translation layer that can span multiple devices. This enables optimizations based upon having visibility across devices not possible from a translation layer embedded in a single device. It also provides greater flexibility, such as maximizing I/O performance for a given application or exploiting specific workload properties.

Radian RMS-350 Zoned Flash SSD

The RMS-350 SSD is based upon Radian's Zoned Namespaces implementation that is targeting compliance with the forthcoming Zoned Namespaces (ZNS) specification from the NVM Express™ industry standards organization. This award winning zoned Flash technology virtualizes the topology of the NAND memory, including geometry and vendor-specific attributes, to present hosts with zones of Idealized Flash while maintaining symmetric alignment through to the physical memory. Comprised of NAND Erase Units (blocks), these zones are subsets of physically, performance isolated regions of memory and are user configurable.

Radian's Address Space Layout (ASL) Configurator enables user configuration of different size zones, write stripes, iso-regions, and iso-boxes. In addition to being based upon capacity, these isolated regions can



be configured based upon endurance, I/O bandwidth, predictable I/O latency, cleaning policies, deterministic scheduling and other metrics via parameterized descriptions.

This configurability helps address the challenges of complex scheduling, along with the sizing tradeoffs associated with zones and write stripes. It also minimizes modifications to host system software and associated integration efforts, while eliminating the write amplification that would otherwise occur if the host segments and SSD zones were not aligned.

As the default option, Radian Zoned Flash SSDs perform *Decoupled* wear leveling and *Decoupled* NAND maintenance. When the host performs data movement as part of its cleaning process, the Radian Zoned SSD can internally, concurrently perform wear leveling and data scrubbing on the applicable zones in a coherently aligned manner that does not interfere with other host-directed I/O accesses. If this management is inadequate, the SSD can use its *Back Channel* out of band communication path to initiate a corresponding request to the host.

Configuration and Integration

The SALSA integration examined in this case study utilized a single Radian Zoned Flash SSD. A subsequent, live demonstration involved two instances of SALSA, each operating on separate processors and accessing six zoned SSDs in dual port mode.

A ZBD-to-NVMe bridge was supplied by Radian to provide a protocol translation from the zone block device interface to NVMe.

As previously discussed, the Radian ASL Configurator provides configurations for different size zones, iso-regions, write stripes and iso-boxes comprised of one or more iso-regions that can be attached to namespaces. Two different drive settings were configured for the case study. The first involved configuring the drive as one iso-box and one namespace, with 62 iso-regions inside that one iso-box. The second configuration involved creating four iso-boxes, with 15 iso-regions inside each iso-box, and assigning a namespace to each of the four iso-boxes.

The zones, comprised of NAND Erase Units (blocks) from within the same iso-region, appear as a range of contiguous LBAs accessible via conventional addressing.

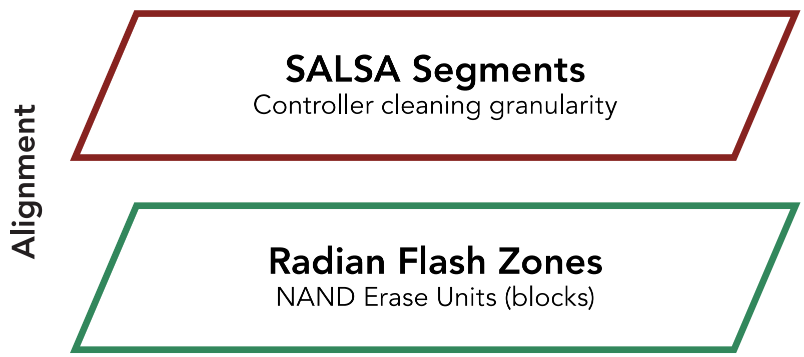
The zones in the Radian SSD were configured to the smallest size while preserving quad plane access.

SALSA issued a 'Zone Report' command to the RMS-350 SSD to discover the device topology.

At initialization SALSA's SCM creates pools of storage and an address space, and the SALSA controllers can subsequently provision their capacity through requests to the SCM.

For the first configuration in this case study, SALSA combined multiple iso-regions into a single address space and namespace. A single SALSA controller was used in this first configuration, and zones were assigned to it by an allocator in the SCM.

SALSA supports multiple frontends and a Linux kernel device-mapper frontend was used in this case study. The SALSA controller exposes a block device and maintains a mapping between user-visible LBAs and backend PBAs referred to as Log Structured Array (LSA) controllers.



Matching SALSA segment size and Radian SSD zone size provided the lowest possible system level Write Amplification

Zone Excursions

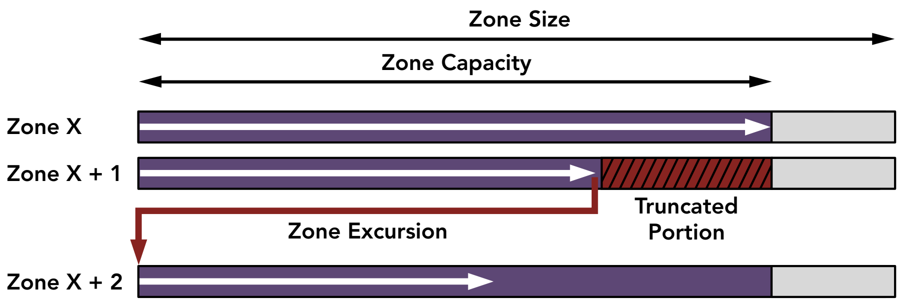
A concept being considered around zoned Flash is the idea of variable zone sizes that allow zone capacity to change over time. NAND Erase Units (blocks) will experience failures over the life of a SSD. Rather than have the drive transparently manage these failures, variable zones would have a size that stays fixed over the life of the drive, but the capacity of storage in a given zone could shrink over time.

One of the potential scenarios around variable size zones involves a host writing across a zone frontier where a sudden write error occurs before the advertised zone capacity has been utilized, yet the zone which has now on its own accord reduced its capacity, rejects any additional writes to that zone. This forces what is called a zone excursion, where the host must jump to a different zone to continue its write stream. Hosts would not only need to handle the impact of these sudden excursions, but they would also need to contend with different zones becoming different capacities, and the gaps between zones becoming different distributions, and then create write stripes across those asymmetric zones and addressing gaps, sometimes doing so dynamically.

Zone Append

A condition known as tangled ordering arises when host software issues write requests sequentially, but the data arrives at the zoned storage device out of order due to chipset anomalies or packet reordering attributed to transport or fabric protocols. To overcome tangled ordering, other zoned FlashSSDs either require that the zones are only accessed at single queue depth, or implement a new command called Zone Append. Operating in single queue depth results in low performance, while implementing Zone Append is typically a major development effort and can be incompatible with many data protection or availability implementations (RAID, Snapshots, etc.).

When managing sequential zones with host-managed SMR hard drives, SALSA forces allocated pages to be written sequentially to the drive to avoid out of order write errors. SALSA does so via one I/O thread per open SMR zone that ensures that all writes to each open zone occur in-order which, while acceptable for SMR hard drives, would impede performance with SSDs. This is not required for FTL-based SSDs, where the SALSA I/O threads are not utilized, but may be required for certain zoned Flash SSDs.



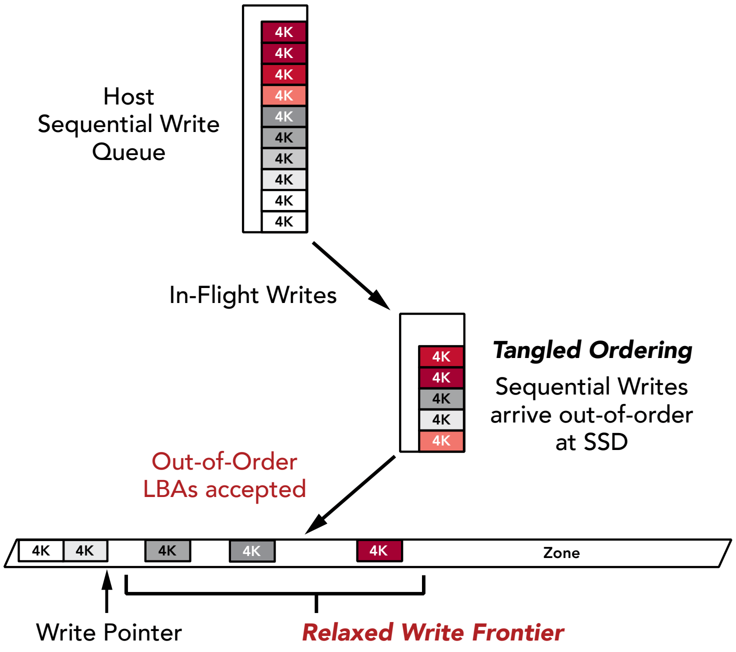
Radian's Idealized Flash prevents zone capacities from changing during operation, so that hosts do not need to manage scenarios like zone excursions or variable address gaps

This is untenable for most data center storage software systems, but was not a challenge for IBM Research in this integration because of Radian's Idealized Flash. In addition to abstracting low level NAND attributes and vendor-specific anomalies, Idealized Flash and ASL prevent zone capacities from changing over the life of the drive, and consequently prevent zone excursions.

This I/O thread was also not required with the RMS-350 SSD because of its Relaxed Write Pointer capability. Enabled by Radian's Idealized Flash, the RMS-350 zones support LBAs arriving out of order to the same zone even at higher queue depth and with multiple concurrent open zones. In addition to being the highest performance option, this capability was fundamental in enabling SALSA's rapid integration of a zoned Flash SSD.

Radian's Relaxed Write Pointer capability enabled SALSA to achieve significant performance gains with minimal modifications.

Alternative approaches, such as single queue depth, would yield poor performance while implementing Zone Append would have been a major software development effort.



Garbage Collection

With zoned Flash SSDs, the host controls garbage collection. This involves selecting zones for free space reclamation along with relocating valid data and determining the new destination location.

For the first configuration in this case study, SALSA's SCM executed the algorithm that selects the zones it deemed best for relocation and notified the one owning SALSA controller to relocate the valid data to a different zone. SALSA uses two configurable water-marks (low or high) to start or stop garbage collection. The SCM is not aware of which blocks are valid. Once the SALSA controller relocates the associated valid data, it frees the corresponding blocks, and zones are eventually freed by issuing a 'Zone Reset' command to the SSD at which point they are returned to their corresponding free queues.

Wear Leveling & NAND Maintenance

With conventional FTL-based SSDs, the wear leveling process is typically integrated with the garbage collection processes and algorithms. With zoned Flash drives, the host is in control of garbage collection.

While many host architectures already perform cleaning that can be directly applied to Flash garbage collection, these host architectures do not possess the logic to perform comprehensive wear leveling, such as accounting for instances of wear imbalance, or NAND maintenance that can be very unique to the specific Flash memory.

Because SALSA is log structured and in control of the zoned Flash SSD's free space, writes will inherently tend to level wear and address many data retention requirements. If this relocation data movement is inadequate, the Radian Zoned Flash SSD's Decoupled wear leveling and NAND maintenance will determine that these processes are required and can perform them in a deterministic, coherently aligned, transparent (abstracted) manner that does not interfere with other host-directed I/O accesses. Similarly, the Radian Zoned SSD performs bad block management by transparently remapping erase units (bad blocks) from erase units held in reserve. This swapping of erase units is again handled deterministically and typically without impacting host latencies.

Performance Comparison

IBM Research performed an 'apples to apples' performance comparison between SALSA and a FTL SSD, and SALSA with the Radian Zoned Flash SSD.

'Apples to Apples' – Identical SSD Silicon

The FTL SSD utilized in this testing was a generally available, enterprise U.2 NVMe SSD from a Tier I vendor. The primary silicon in this FTL SSD is identical to the Radian Zoned Flash SSD. This includes the same SSD processor, the same NAND memory device (type and die stack) laid out in the same array configuration (channels and number of NAND devices/channel), and the same DDR memory device laid out in the same array configuration (number of devices and ECC). As a result, the total raw Flash capacity and DDR capacity are identical between the two SSDs.

The test platform utilized for each comparison was a Lenovo® System x3650 M5 model 5462WFX, with two Intel® Xeon® CPU E5-2630 v3 @ 2.40GHz, and 160GB DRAM with a Linux 4.15.16 kernel.

System Level Testing

Tests were performed with SALSA and each SSD using the fio block level tester and the SysBench database tester utilizing mixed random read/write workloads. Different apples to apples system level comparisons were configured to examine effects such as log-on-log write amplification and configurable isolation (noisy neighbors), including isolation for applications.

System Level Overprovisioning

The raw Flash capacity of 4.62 TB was the same for each SSD. The total system level overprovisioning of 30% was the same for SALSA with the FTL SSD, and SALSA with the Radian Zoned SSD, and each configuration provided the same advertised user capacity of 3.3TB.

Like most advanced storage management layers, such as found in All-Flash Arrays, SDS frameworks and hyperconverged architectures, SALSA requires free space for cleaning and space reclamation that requires overprovisioning of storage capacity. This is not unique to Flash media, and is also the case for most modern purpose-built storage management layers when utilizing hard disk drives.

FTL-based SSDs also require device level, internal overprovisioning for free space for internal garbage collection, but this is not required by the Radian Zoned Flash SSD. In this testing, while 30% of the total Flash capacity is overprovisioned for each SSD, the FTL SSD internally consumes over 17% of the capacity, leaving the SALSA host management layer with only 13% free space. Utilizing the Radian Zoned Flash SSD enables approximately doubling the SALSA free space while maintaining the same 30% total system level overprovisioning. It also removes the write amplification from either the 'log-on-log' effect or full stripe write approach that would otherwise occur with a FTL-based SSD.

Configuration

Both the FTL SSD and Radian Zoned Flash SSD present themselves as NVMe block devices. The Radian device, through the aforementioned NVMe-to-ZBD bridge, provides zones and logical blocks to SALSA while the FTL SSD only provides logical blocks. Due to time and resource constraints, the SALSA integration did not utilize the Radian Delegated Copy-Move functionality.

The performance evaluation involves three different ‘apples to apples’ comparisons. The configurations for each comparison involve a single instance of the SALSA storage management layer and a single SSD.

Comparison 1 – fio, single job

This comparison utilizes the fio tester generating a 50% 4K random read/50% 4K random write workload to SALSA. With both the FTL SSD and the Radian Zoned Flash SSD, SALSA generates four threads with queue depth = 16/thread, total SSD I/O Depth = 64.

Comparison 2 – fio, four jobs

In this comparison, SALSA configures four LSA controllers instead of one LSA controller. The FTL SSD is accessed as four separate namespaces, while the Radian SSD is accessed as four separate iso- regions. Each iso-region could have also been designated with a unique namespace but that was unnecessary for these testing purposes. Otherwise, the configuration for Comparison 2 is largely the same as Comparison 1. One fio job is assigned to each of the four LSA controllers. SALSA creates threads to serve SYNC writes for each respective zone, using a max write queue depth of 4 per region, for a total device write I/O depth of 256. Non-sync writes to zones also adhere to the same maximum I/O depth but are issued directly from the host thread context. Reads are served directly from the host thread context without SALSA-level I/O depth limits; the same is true for writes to non-zoned devices. The same 50% 4K random read/50% 4K random write workload is generated by fio.

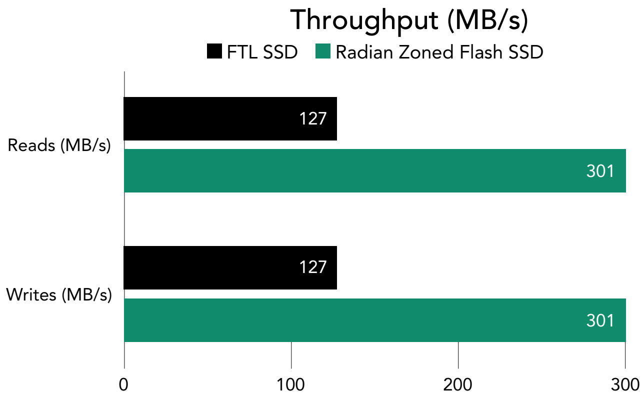
Comparison 3 – fio and MySQL/SysBench

The SSDs in Comparison 3 were configured with four separate namespaces or iso-regions, the same as they were in Comparison 2. Three SALSA LSA controllers, each accessing a separate namespace or iso-region, serviced an fio job, each with a single thread at queue depth of 16/thread, the same as in Comparison 2 with the same aforementioned fio workload.

The difference in Comparison 3 is with respect to the first namespace or iso-region where, instead of fio, the SALSA LSA controller associated with that region supported an instance of MySQL hosted in a docker® container. Sysbench was used as the database tester and generated a default OLTP workload, using 16 threads, 64 tables, and 10M entries/table.

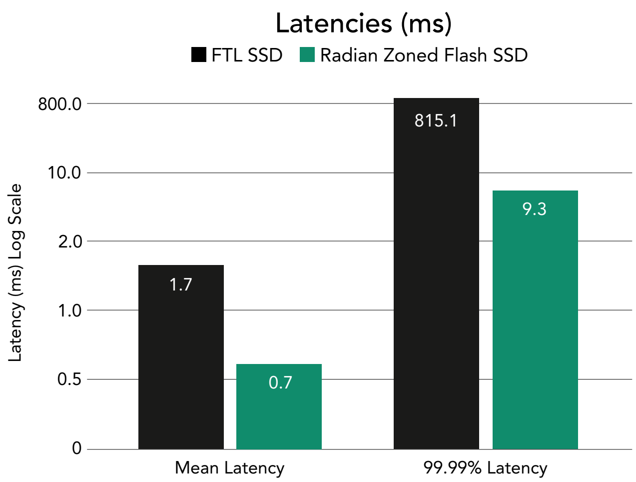
Comparison 1

fio, single job, four threads, QD = 16/thread, IOD = 64
50% 4K Random Read/50% 4K Random Write



Over 2x improvement

Data center storage management layers such as SALSA (and as found in most All-Flash Arrays, SDS frameworks and hyperconverged architectures), require their own overprovisioning as a fixed cost, even with hard drive media. FTL SSDs also require internal overprovisioning. This redundancy creates inefficient resource utilization that results in suboptimal performance at the system level, especially with mixed workloads.



Write bandwidth is a significant factor in determining overall system latency. In this mixed workload, the FTL SSD only provides 127 MB/s of system level write bandwidth. The internal traffic appears to reach an inflection point, even at this relatively low write bandwidth, that creates very large system level tail latencies in comparison to the Radian zoned SSD.

Write Amplification Factor (WAF)

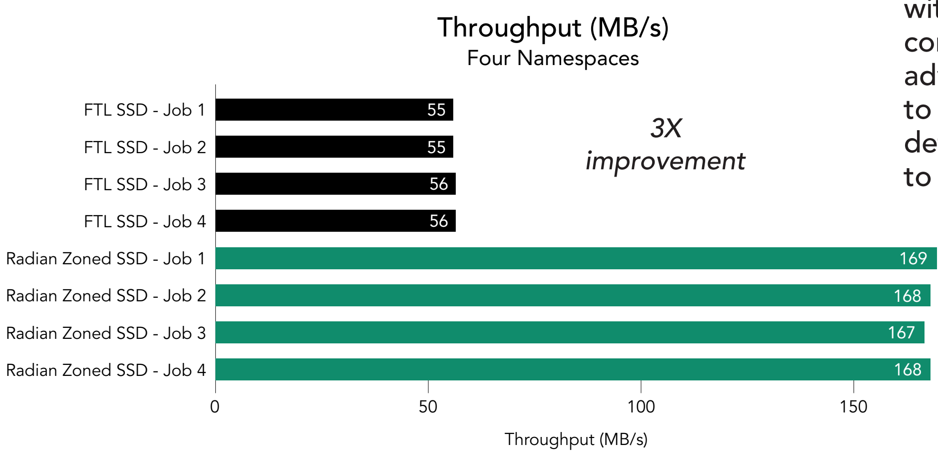
Write amp. is greatly reduced with the Radian SSD because zone erase segments can be configured to align with system space reclamation. In addition to improving SSD endurance, this reduction in write amp. dramatically improves latencies.

Aside from zone and write stripe alignment, overprovisioning will determine the WAF. Knowing the raw and advertised capacities of each SSD under a random workload, a rough estimated approximation based upon the paper, "Analytic Models of SSD Write Performance", Desnoyers[1], indicates a *host level* WAF of at least 2.6 for the Radian SSD, and at least 3.8 for the FTL SSD (46% higher) in this benchmark comparison. This is a result of the larger free space being allocated to SALSA with the Radian SSD, even though the total system overprovisioning is the same as when SALSA utilizes the FTL SSD.

However, in addition to host side write amp., FTL SSDs also have internal *device level* write amp, and the *system* WAF is a multiplicative product of the host WAF and device WAF. If we conservatively assume that the FTL SSD has an internal device WAF of 2.0, the system level WAF would be 7.6 (2.0 x 3.8). This is almost 3x the estimated 2.6 system level WAF with the Radian SSD because, in addition to providing the SALSA host with additional free space, the Radian Zoned SSD does not have internal device level write amp.

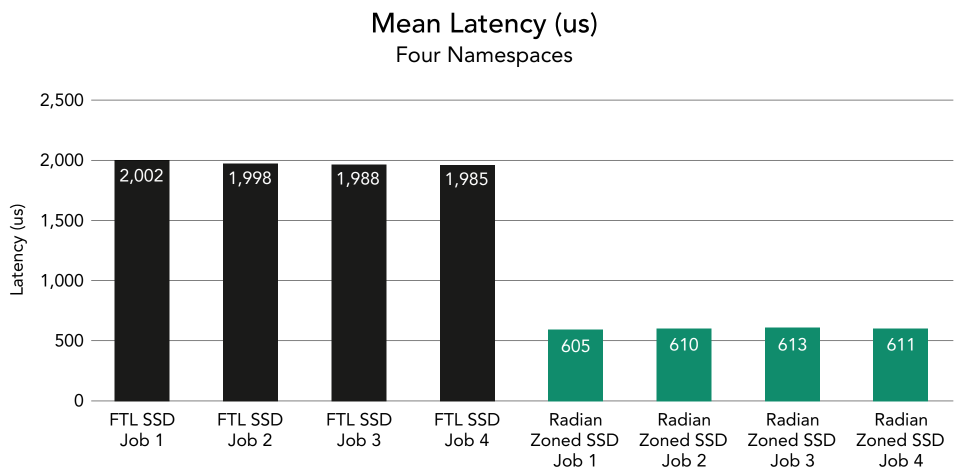
Comparison 2

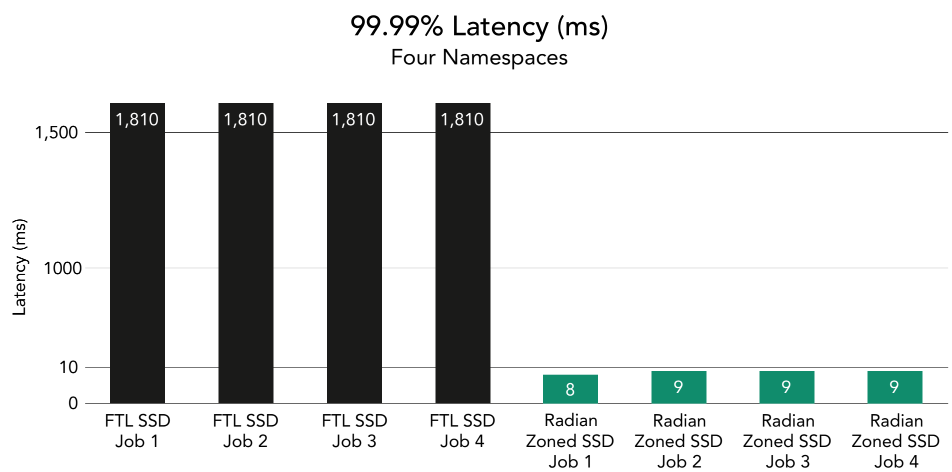
fio, four jobs, four threads, QD = 16/thread, IOD = 64
50% 4K Random Read/50% 4K Random Write



SALSA's flexible architecture with variable instances of LSA controllers can readily take advantage of the Radian SSD to concurrently provide deterministic performance to different clients

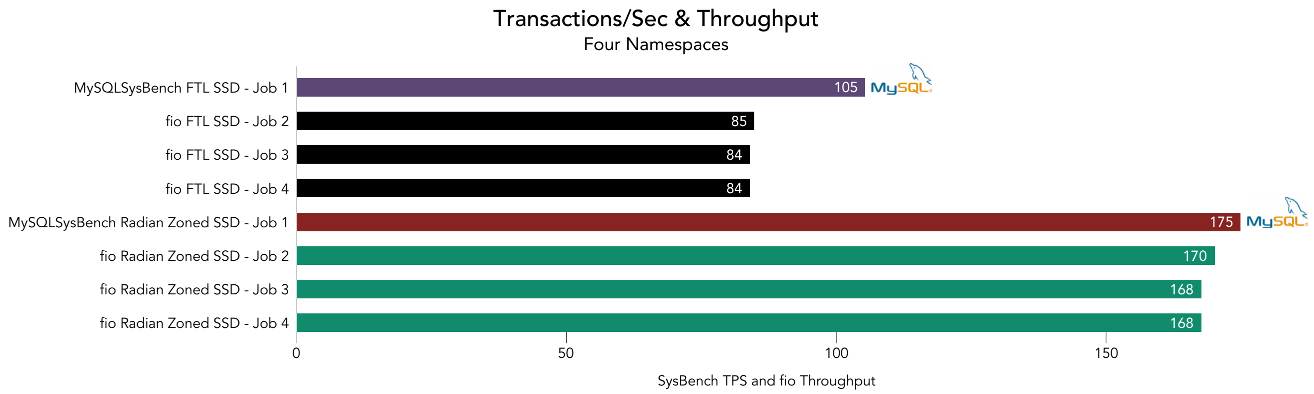
Both the FTL SSD and the Radian Zoned Flash SSD provide support for multiple namespaces. However, the Radian Zoned SSD also provides physically isolated regions that, along with zone (segment) size and write stripes, are configurable. This combination of isolation and configurability simultaneously improves throughput, latency, and determinism.



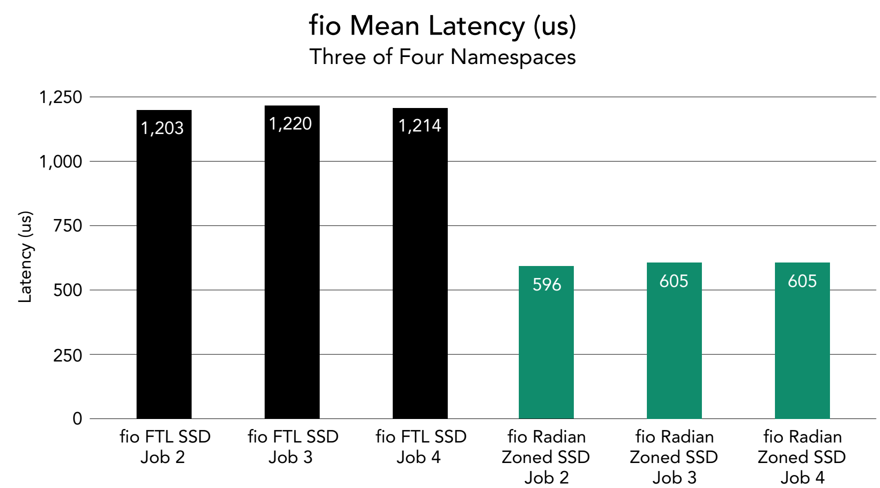


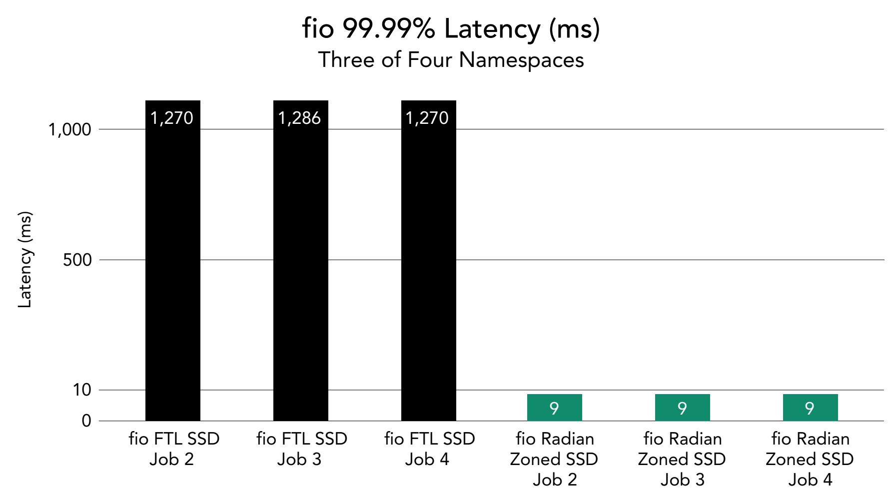
Comparison 3

SysBench, one job, 16 threads, OLTP workload with 64 tables, and 10M entries/table
fio, three jobs, three threads, QD = 16/thread
50% 4K Random Read/50% 4K Random Write

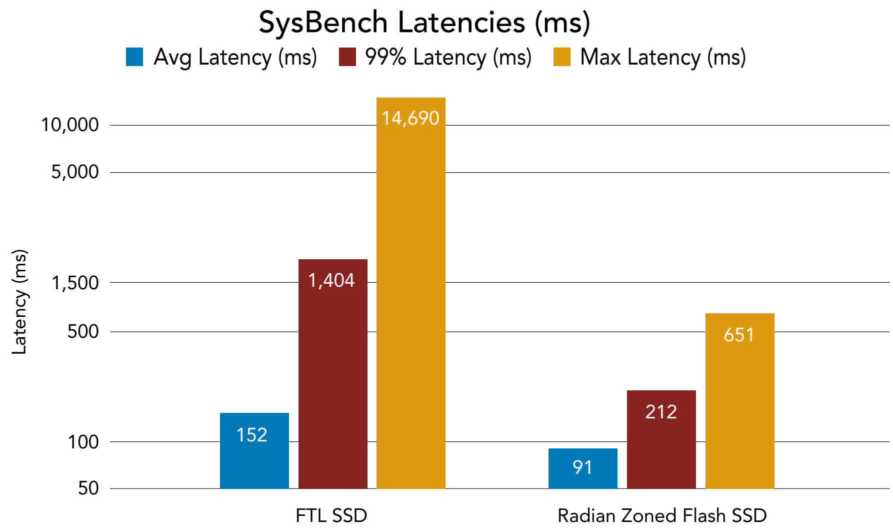


This configuration illustrates how the combination of SALSA and the Radian Zoned Flash SSD translates into system level efficiencies for applications. Configuring write stripes to match client applications improves parallelism, which plays a significant role in improving TPS and throughput. In this example, MySQL TPS under SysBench improved by 66%.



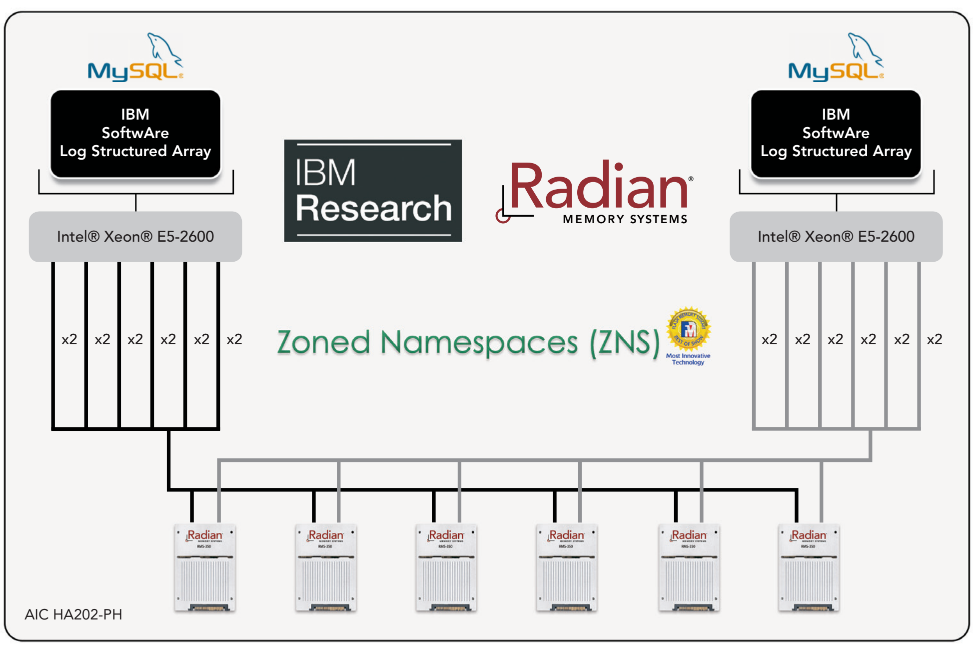


While cross traffic from adjacent clients (the three fio jobs in this example) would normally create interference for the MySQL client, SALSA instead allocates each client to a dedicated isolated region. In addition to superior latency profiles across different metrics, this performance can scale predictably with additional clients and capacity.



Dual Head Multi-Drive Demonstration

In addition to the performance testing performed by IBM Research, Radian also provided a demonstration of SALSA with multiple Radian Zoned SSDs in a high availability configuration at the USENIX File and Storage Technology (FAST) conference held in February, 2020. The demonstration involved two instances of SALSA with six RMS-350 4TB zoned SSDs. Each instance of SALSA accessed one port and 2TB of each of the six zoned SSDs representing a single 12TB volume. Testing involved fio as a single tester, multiple fio testers (noisy neighbors), and SysBench with MySQL, with the respective tests running concurrently on each processor.



Visit the Radian Memory Systems®, Inc. booth to see a live demonstration of IBM® Research's SoftwAre Log Structured Array storage management stack [1] running MySQL® with Zoned Flash SSDs in a high availability configuration

[1] Ioannou N., Kourtis K., and Koltsidas I. .Elevating Commodity Storage with the SALSA Host Translation Layer. MASCOTS'18

Summary

Today's data centers predominantly access SSDs as shared storage resources, which requires a storage management layer. Like most advanced All-Flash Arrays, purpose-built SAN and NAS appliances, hyper-converged and SDS frameworks, the SALSA storage management layer is non-overwriting to achieve scalable performance and efficient space management. This also makes it an excellent candidate for zoned Flash SSDs.

More specifically, as SDS frameworks continue to gain momentum throughout data centers, a requirement arises for a versatile, local backend storage engine. SALSA fulfills this requirement with its ability to easily match different access patterns with efficient local space management while still providing SDS frameworks and distributed filesystems with a generic block interface.

With respect to providing direct support for local applications, the SALSA architecture, in combination with Radian's ASL functionality, exploited the parallelism of the Radian SSD up to the application level as illustrated in the comparative benchmark testing. This enabled a standard, unmodified database to achieve dramatic improvements in speed with orders of magnitude improvement in tail latencies over the same stack utilizing a conventional SSD. In a real world deployment, these advantages would translate into providing clients with superior performance while simultaneously reducing TCO, without any modifications to the applications.

Down at the block level, under fio tests, SALSA achieved remarkable improvements of 3x in throughput with 50x improvements in tail latencies with the Radian SSD in comparison to using a FTL SSD that had identical silicon.

Just as importantly, write amplification was estimated to improve by almost 3x under both the fio block level tests and SysBench database application tests. This improvement not only translates directly into improving Flash wear out and useable product life, but even enables new, lower endurance memory technologies such as QLC, and for a wider range of potential applications.

While achieving dramatic performance and write amplification improvements with Software-Defined Flash SSDs is becoming more readily understood, these performance results often come at a steep cost with respect to host software modifications and associated development efforts.

This is also true of many forthcoming zoned Flash SSD architectures, where system designers will be forced to choose between running in a single queue depth mode per zone for writes, with low associated performance, or undergoing a major redevelopment effort implementing support for a zone append model and the associated design and testing implications for existing fault tolerance mechanisms.

With the Radian RMS-350 Zoned Flash SSD, IBM Research did not need to make that choice. The Radian SSD's Idealized Flash precluded the requirement to run in single queue depth mode, and hence obviated the requirement to implement zone append, while achieving superior system performance than either of the alternative approaches would have provided.

Similarly, the SALSA storage engine did not need to take on the responsibilities, or undergo what would be significant modifications, to account for low level NAND attributes, gaps in addressing, variable zone capacities and excursions, implementing logic for NAND maintenance or wear leveling corner cases.

In conclusion, IBM Research was able to rapidly realize significant performance gains and endurance improvements with the Radian Zoned Flash SSD with relatively minor software modifications and a rapid integration process that only required several man weeks of engineering time.

SALSA, combined with the Radian Zoned Flash SSD, provides a highly efficient, performant backend storage engine for SDS frameworks or as a directly accessible shared resource capable of providing deterministic, scalable storage to multiple applications running concurrently.

About IBM Research

IBM Research is one of the world's largest and most influential corporate research labs, with more than 3,000 researchers in 19 locations across six continents. We play the long game, investing now in tomorrow's breakthroughs. Watson, the world's first cognitive system, is the fruit of over 50 years of IBM research in artificial intelligence. Today, it forms a core part of IBM's business.

Our scientists are charting the future of artificial intelligence, breakthroughs like quantum computing, how blockchain will reshape the enterprise and much more. We are dedicated to applying AI and science to industry challenges, whether it's discovering a new way for doctors to help patients, teaming with environmentalists to clean up our waterways or enabling retailers to personalize customer service.
www.research.ibm.com

About Radian Memory Systems

Radian has pioneered Cooperative Flash Management, a new paradigm that redistributes Flash management responsibilities between the host and device to achieve superior efficiencies across performance, cost, and endurance metrics. This has included shipping the first zoned Flash device and creating more Software-Defined Flash innovations than any other company in the industry. Radian's technology and products target system OEMs, cloud and service providers, and licensing to device-based manufacturers to support primary storage requirements throughout the data center.
www.radianmemory.com

[1] "Analytic Models of SSD Write Performance", Desnoyers P., Northeastern University, ACM Transactions on Storage, Vol. 10, No. 2, Article 8, Publication date: March 2014. (https://dl.acm.org/doi/10.1145/2577384)