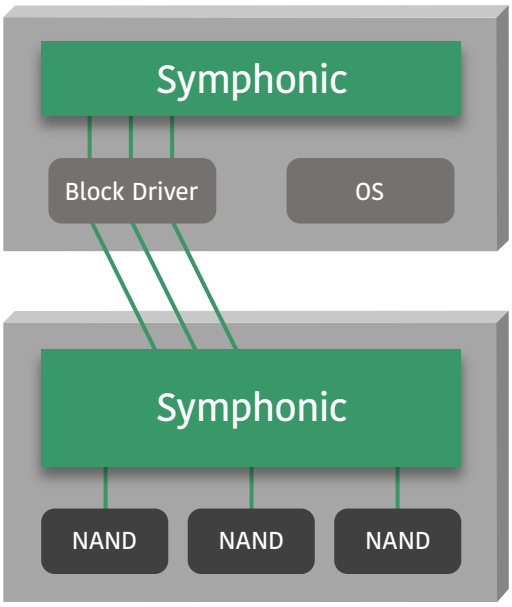
Symphonic™

Cooperative Flash Management

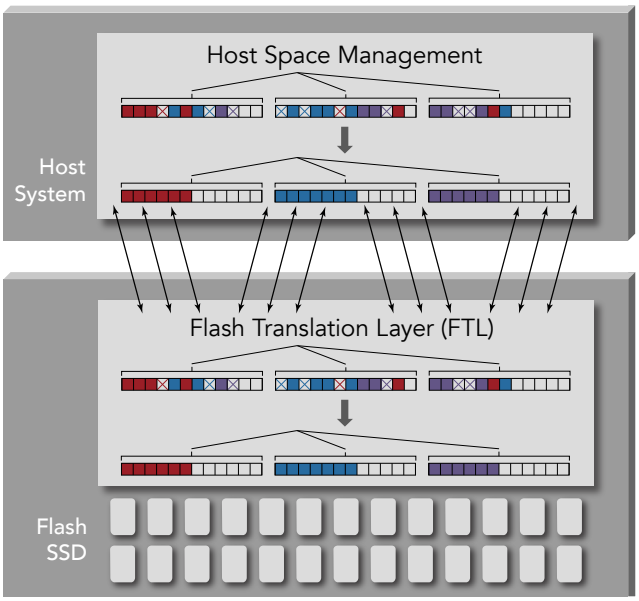


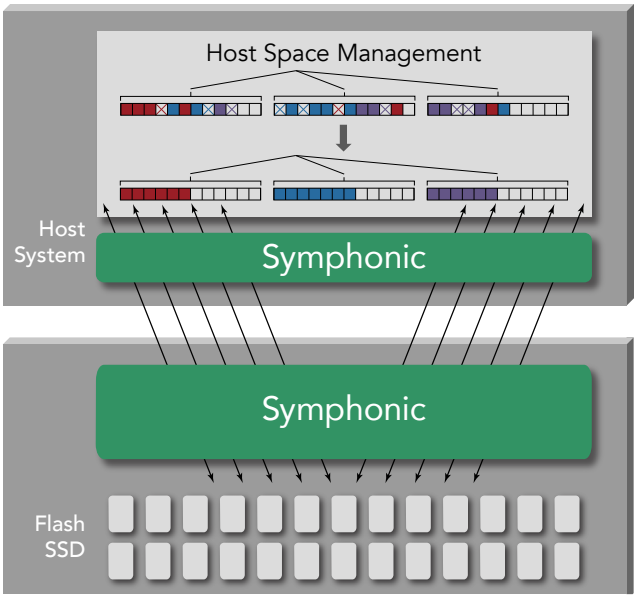
Redefining Software for Flash Storage

Radian®
MEMORY SYSTEMS

Unlocking native Flash performance

By taking a system-driven approach, Symphonic redfines software for Flash storage. Replacing the SSD Flash-Translation-Layer (FTL), Symphonic overcomes performance, cost, and endurance limitations while providing the functionality required of a data center class product.





FTL SSD

- Redundant levels of space management, lookup and locking penalties
- Two levels of write amplification
- Dislocation from two asynchronous processes
- Barrier to realizing host intelligence, data locality and layout optimizations on media and across storage volumes
- Requires additional Flash capacity overprovisioning

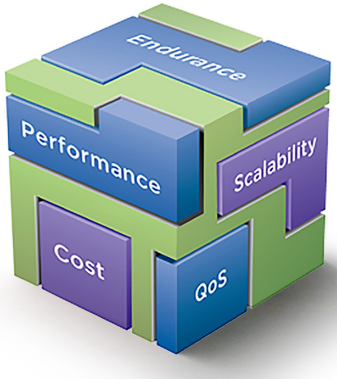
Symphonic SSD

- Single level of space management, lookup and locking penalties
- One level of write amplification
- Management processes are coherent
- Host's global intelligence, scheduling, knowledge of data locality and layout optimizations realized on media and across storage volumes
- Minimizes capacity overprovisioning

Cooperative Flash Management

Symphonic enables system software, e.g., file systems, block virtualization managers, or object/key value stores to cooperatively perform Flash management processes.

Solving Software-Defined Flash



Enabling a New Era for Flash in the Data Center

Next Gen data centers require highly deterministic, low latency and cost effective storage that can only be realized with Flash memory. But every Flash SSD solution available today is based upon a Flash-Translation-Layer (FTL) that introduces expensive overhead, unpredictable latency spikes, and suboptimal performance that prematurely wears out the Flash media.

Symphonic is a type of Software-Defined Flash that creates a new paradigm for Flash memory management that we call Cooperative Flash Management (CFM). System software, e.g., file systems, block virtualization managers, or object/key value stores have comprehensive capabilities to intelligently manage storage media. But this system software is not equipped to directly perform some of the unique processes required to manage Flash memory.

Replacing the SSD FTL, Symphonic includes a combination of host-side software libraries and SSD firmware that enables system software to cooperatively perform Flash management processes to realize the full potential of Flash storage. The Symphonic functionality includes configurable address mapping, garbage collection, wear leveling, and reliability features that turn the SSD into an offload engine while operating in host address space. The result is a redistribution of host/device responsibilities that removes an inefficient abstraction layer to dramatically improve Quality-of-Service, performance, cost, and endurance while providing the functionality of a data center class product.

>80% increase in IOPS & User Bandwidth
By operating in host address space, Symphonic enables greater system parallelization, optimized data layout, and avoids duplicate table look ups and locking penalties.

Magnitude improvement in QoS latency
Providing geometric alignment from the host down to the 'bare metal' avoids collisions which are the primary source of unpredictable latency spikes. Combined with performing garbage collection under intelligent host control, Symphonic provides a magnitude improvement in latency QoS that cannot be achieved with FTL architectures.

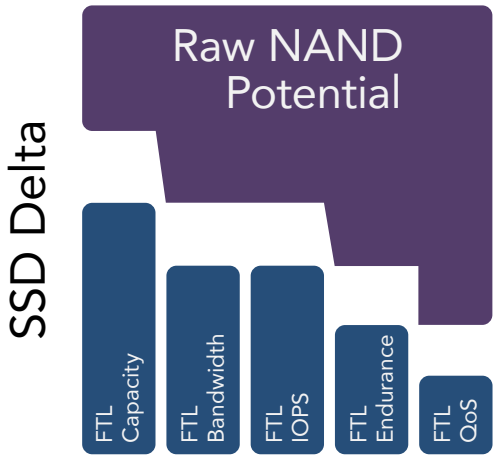
Metrics scale linearly with additional SSDs
Symphonic turns the SSD into an offload accelerator engine, freeing host resources while enabling linear scalability of IOPS/bandwidth, latency/QoS, and CPU/I-O as SSDs are added to a system.

Cost Savings – Raw Flash reduced 15% or more
Symphonic does not require additional overprovisioning of raw Flash capacity for processes such as garbage collection. This results in reducing overprovisioning requirements from the 10% to 30% common in enterprise FTL SSDs down to less than 3%, in addition to reducing system-level overprovisioning.

Endurance – Write Amp. factor reduced 75%
By operating in host address space and performing Flash management processes under cooperative host control, Symphonic avoids the additional level of device write amplification induced by FTL SSDs to dramatically increase the usable life of the Flash media.

Software-Defined Flash:
the answer to the SSD Delta

Why can't conventional SSDs deliver the capacity, performance, or endurance of their own raw Flash?



FTLs and the SSD Delta

Utilizing NAND memory for data storage requires performing Flash management processes, such as garbage collection and wear leveling.

SSDs perform these processes transparently in a black box known as the Flash-Translation-Layer (FTL) that abstracts the processes and emulates hard drives for backwards compatibility. But this abstraction significantly reduces the native performance and cost metrics available from the raw NAND memory, creating a gap that we call the SSD Delta.

Software-Defined Flash

Software-Defined Flash (SDF) removes the FTL, enabling the host software stack to directly perform Flash management processes and realize the full potential of the raw NAND memory.

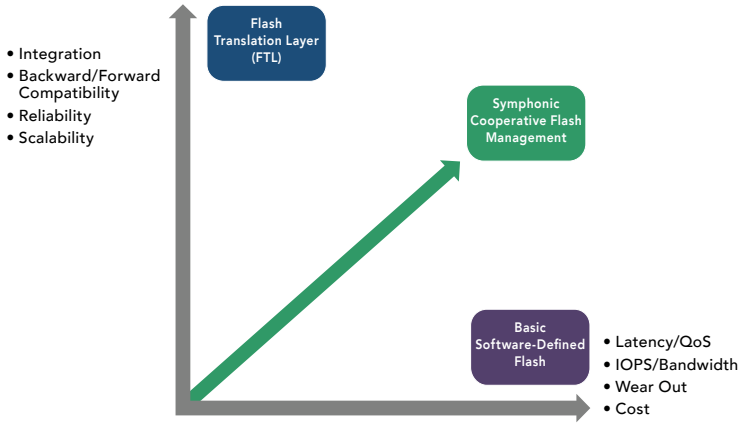
But in removing the FTL, other forms of SDF also create a new set of technical challenges.

Challenges from SDF

- Difficult integration and significant modifications to target host software stacks
- Burdens host with low level NAND details
- Lacks Reliability, Availability, and Serviceability features
- Lacks Forward Compatibility and ties host software to vendor-specific NAND properties
- Prevents SSD suppliers from including micro-op endurance optimizations and supporting product warranties
- Poor scalability as SSDs are added to a system

Symphonic

As a subset of SDF, Radian's Symphonic technology preserves the SDF advantages while specifically addressing each of the resulting challenges to provide the functionality required of a data center class product



Symphonic™

Cooperative Flash Management

Offload Accelerator Engine

- Symphonic firmware turns the SSD into an offload accelerator engine
- State machines integrated with mapping and translation tables to perform operations and generate/maintain comprehensive metadata
- Counters track valid/release states with firmware performing heuristic analysis to provide fine grain qualitative range fragmentation metrics
- Provides linear scalability across metrics as additional SSDs are added to the system

Cooperative Garbage Collection

- Garbage collection (copy/move/erase) operations executed by Symphonic firmware, performed entirely on SSD
- Processes scheduled and controlled by host and performed in host address space
- Eliminates additional system copying and overhead, freeing host resources

Wear Leveling & Reliability

- Wear leveling and bad block management performed by Symphonic firmware on the SSD with ECC
- Processes integrated with garbage collection,transparently to the host
- Optional mode to support global host-based wear leveling
- Fault tolerance and support for ACID requirements
- Provides RAS capabilities such as hot swap and FRU functionality
- Each mode enables vendor supported warranties

Geometry Emulation

- Virtualizes topology of NAND array, exporting an emulated version of the physical device geometry to the host to maintain symmetric alignment from host through physical memory
- Minimizes collisions and unpredictable latency spikes while providing maximum parallelization
- Releases host from assuming responsibility for low level NAND constraints
- Forward Compatibility to help 'future proof' system software from evolutionary changes in geometry and vendor-specific NAND attributes

Configurable Addressing

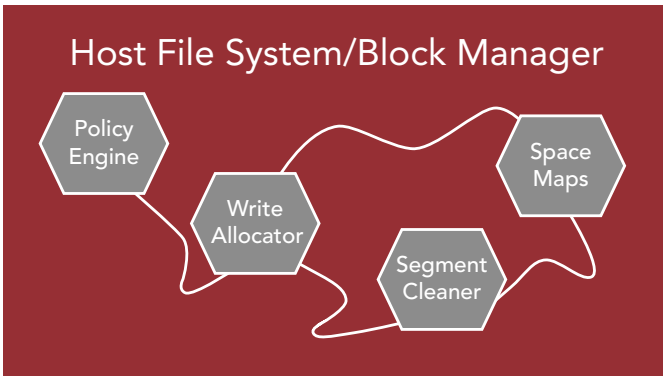
- Symphonic libraries provide extensible connectivity from host space management through the system to SSD firmware
- Facilitates host integration and minimizes modifications to existing host software
- Enables readily optimizing opposing performance and efficiency constraints unique to Flash memory

Cooperative Flash Management (CFM)

System Architecture

Host system software, e.g., file systems, block-level virtualization managers, or object/key value stores typically have comprehensive space managers, segment cleaners, and schedulers to intelligently manage storage media.

But this system software is not equipped to directly perform some of the unique processes required to manage Flash memory, and in particular Flash garbage collection. Symphonic provides the functionality to enable host system software to cooperatively perform this Flash management.

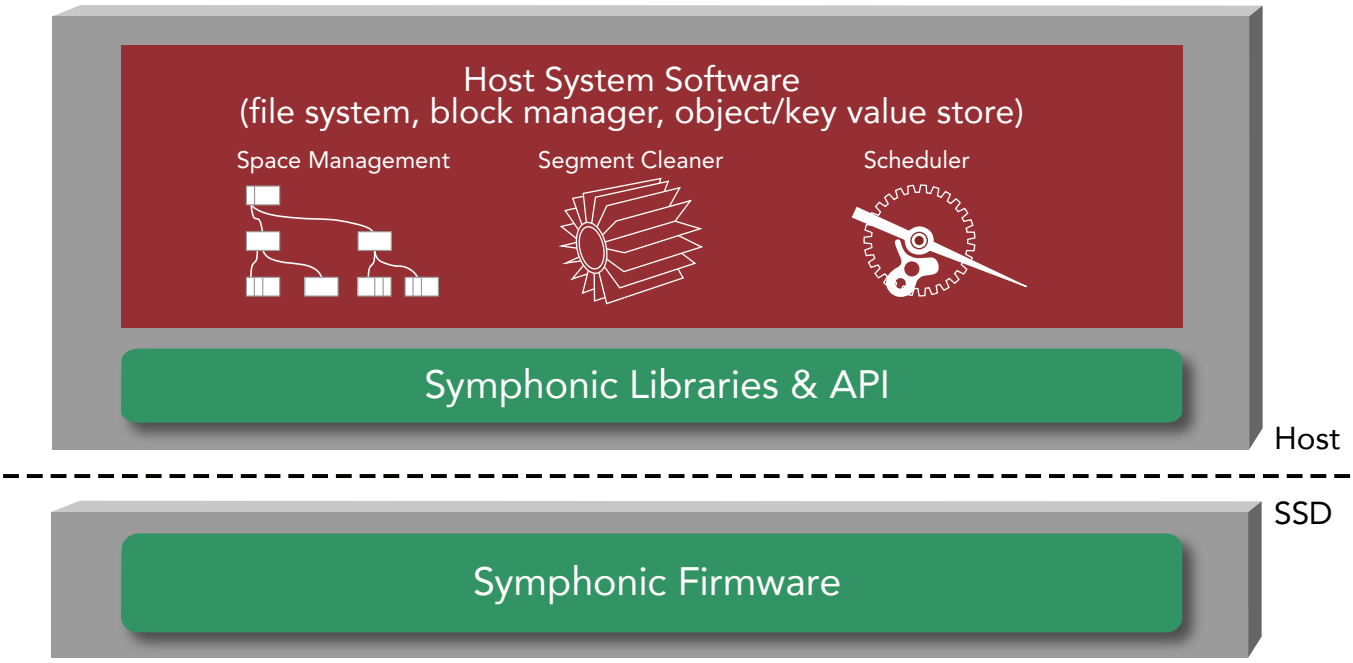


Symphonic followed a system-oriented approach that drove architectural requirements:

- Leverage the host system's intelligence, control and global perspective over workloads, data, prioritization, resources, and load balancing
- Avoid redundancies and overhead
- Realize the full potential of Flash memory while minimizing the modifications to the system's current software stack
- Abstract geometry and vendor specific memory attributes to provide forward compatibility
- Ensure scalability and minimal consumption of host resources
- Minimize wear out and overprovisioning
- Provide a durable solution with data center class RAS functionality

As part of achieving the optimal system architecture, Symphonic redistributes functionality and responsibilities between the host system and SSD. Built on the premise of operating in host address space, Symphonic leverages the intelligent segment cleaning and macro scheduling that the host is already performing. This enables the host to control and schedule processes such as garbage collection, but offloads process execution and lower level media management to the SSD. The Symphonic SSD retains many of the responsibilities of a conventional FTL SSD, in addition to including new functionality to support cooperative control and offload requirements

The right amount of abstraction, at the right level in the stack



Abstracting Flash at the right granularity leverages the host system's extensive capabilities and provides the advantages of operating in host address space.

The Symphonic Address Configurator and Geometry Emulation abstract NAND constraints to minimize host modifications and hardware dependencies while obtaining optimal performance.

Because the SSD is in a superior position to manage NAND programming constraints, media properties and error handling, Symphonic firmware performs these functions along with offloading metadata generation and process execution.

Utilizing NAND memory for storage requires managing lower level, NAND-specific attributes and programming operations. If an existing host system attempted to access a raw NAND storage device directly, it would be unable to achieve the potential performance and efficiencies without significant modifications to the system software and become dependent on very specific SSD hardware. Alternatively, FTLs provide such a heavy and redundant abstraction that the resulting inefficiencies prevent systems from realizing the potential performance and cost advantages from Flash storage.

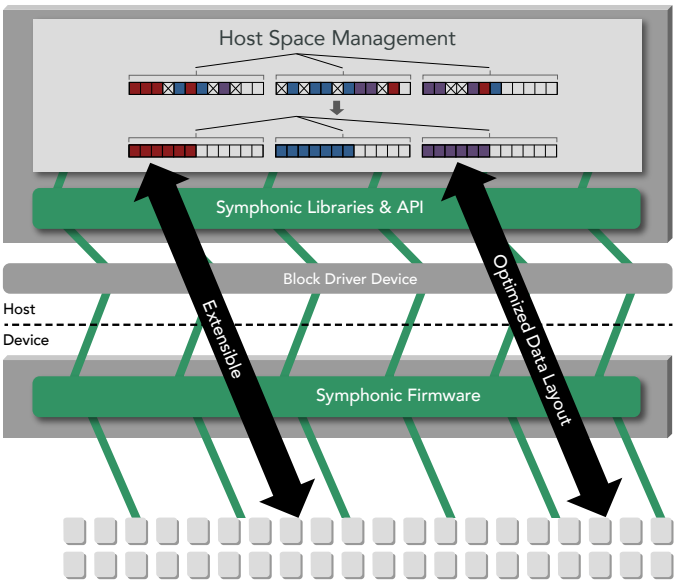
Symphonic Address Configurator & Geometry Emulation

Symphonic's Address Configurator and Geometry Emulation reside in SSD firmware and host libraries. The Address Configurator enables existing host stacks to select the optimal balance between opposing performance and efficiency objectives. Combined with Symphonic's Geometry Emulation, the Address Configurator abstracts NAND constraints and the array's topology while maintaining symmetric alignment with the physical memory array.

The Address Configurator also enables the existing host segment cleaner to control and schedule Flash garbage collection processes. This mechanism enables new, more optimized cleaning policies that cannot be obtained utilizing FTL SSDs. And while the topology of the physical array is abstracted, the host's optimizations around data layout are carried through to placement on the physical media. The result is a solution that minimizes host modifications and hardware dependencies, while enabling control over cheduling and data layout to minimize latency spikes and write amplification, and achieve maximum parallelization.

Symphonic Host Libraries and API

The Symphonic host libraries and API provide extensible access and management, interfacing into the existing functionality of the target host stack to enable cooperative control over the processes executed by the SSD.



Based on standard interfaces with vendor specific extensions, the API and device driver coordinate paths from the host, through the OS, to the Symphonic SSD firmware.

A single layer at the system-level can efficiently span multiple SSDs, avoiding duplication and synchronization of overlapping functionality. This extensible approach provides control and symmetry from the host stack down through the device.

Host intelligence and functionality, such as data locality or TRIM (release), become implicit processes coherent throughout the system.

The Symphonic host libraries and API provide:

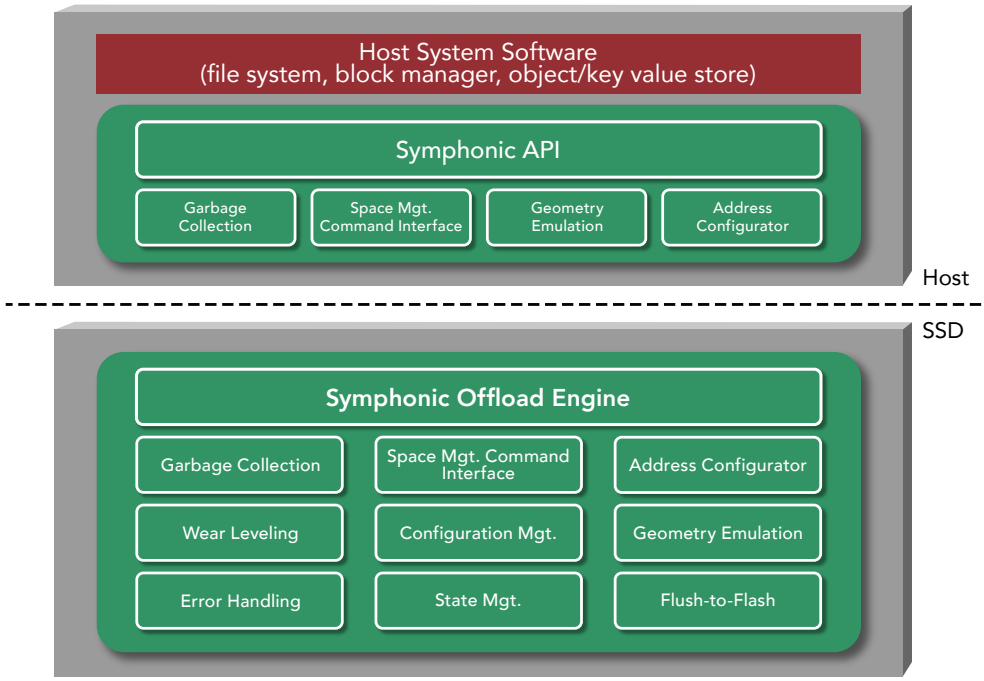
- Interface and mechanisms to the host space management system
- Data accessible via logical block addresses (LBAs)
- Interface for configurable address mapping, layout, and striping with tunable parameters
- Interface for Geometry Emulation that virtualizes physical topology of NAND array while providing symmetric alignment
- Simple and efficient access to formatted metadata generated by Symphonic device firmware
- Commands to control and schedule garbage collection processes
- Optional mode to support host-based wear leveling utilizing metrics generated by Symphonic device firmware
- Mechanisms for setting thresholds, queries, alerts, and event notifications

API & Device Driver

A Symphonic SSD is treated as a block device utilizing an API based on the industry standard NVMe command set, and a design intended to also support SAS interfaces. Initially targeted at the Linux OS, no modifications are required to the mainline kernel and the standard NVMe device driver can be utilized to support most functionality. A modified Symphonic device driver is also available to support specific diagnostic utilities and optional advanced functionality.

Symphonic API:

- Based on the NVMe command set and utilizes the vendor specific management and I/O extensions
- Designed to accommodate SAS (SCSI protocol) utilizing vendor specific extensions
- Portable to different OS environments to support host stacks ranging from block virtualization to file systems and key value/object stores
- SSDs appear as block devices
- Concurrently supports multiple block devices



Symphonic Firmware

The Symphonic firmware is a collection of integrated state machines, mapping and translation tables, and counters explicitly designed to provide CFM. Residing on the SSD's embedded processor, the firmware replaces the FTL and performs on-device garbage collection, wear leveling, and bad block management.

Cooperative Garbage Collection

Cooperative garbage collection is provided by firmware state machines integrated with mapping and translation tables. Statistical counters track valid/release states, with the garbage collection engine monitoring event time stamps and utilizing heuristic analysis to generate fine grain range fragmentation and qualitative metrics. Resulting metadata is consumed by embedded flash management processes in addition to being formatted and exported to host resident libraries, with support for event triggered notifications.

The garbage collection engine can perform copy/move/erase operations entirely on the SSD, under the control of host initiated commands and while operating in host address space. Other modules in the garbage collection engine provide configurable threshold parameters, supporting asynchronous event notifications, queries and alerts.

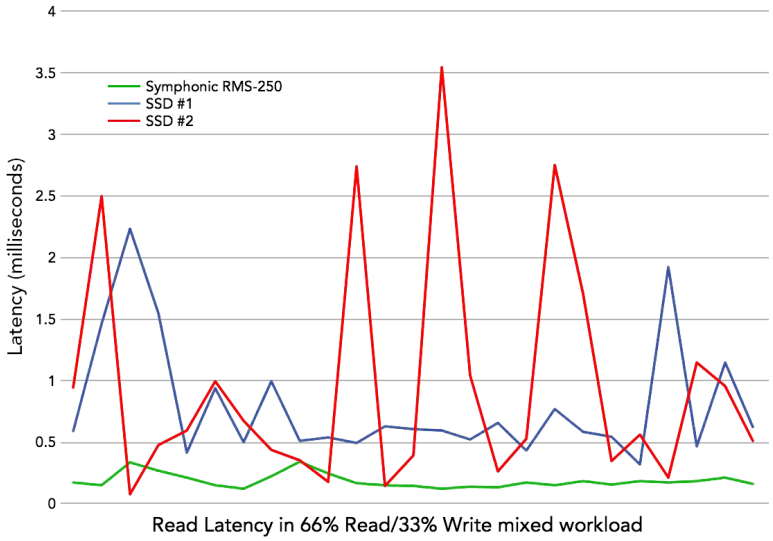
Wear Leveling

Wear leveling and bad block management are handled in a firmware engine that tracks a range of variables accounting for factors such as program/erase cycles, program times, corrected bit errors, retries and disturbs. By default, the wear management functionality is integrated with the garbage collection engine and performed transparently to the host system's segment cleaning process. An optional mode is available to support host-based wear leveling, with each mode implemented in a manner that enables vendor-supported warranties.

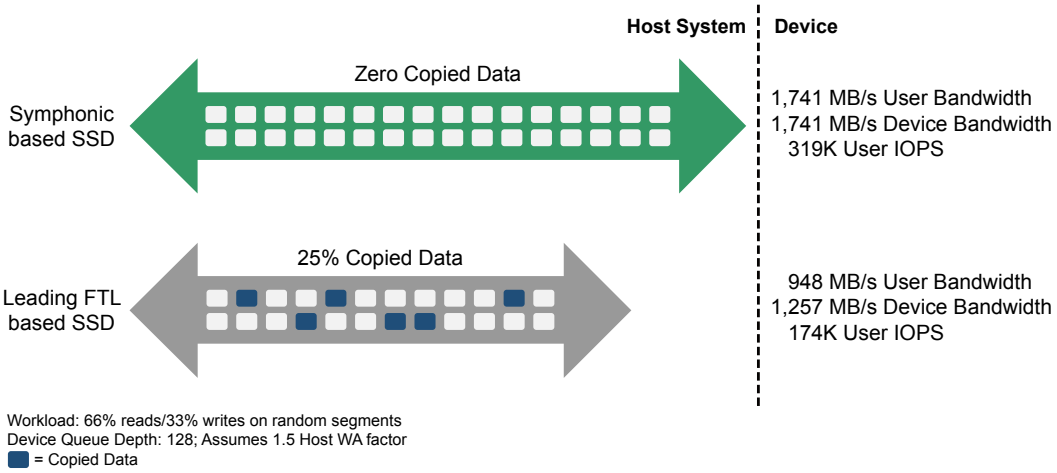
Designed around ACID principles, Symphonic supports RAS (Reliability, Availability, and Serviceability) requirements essential for data center products. This involves fault tolerant maintenance of data and metadata, and capabilities to support treating the SSD as a Field Replacement Unit (FRU).

Geometry Emulation

In exporting the geometry of the device, the Symphonic firmware employs geometry emulation in cooperation with Symphonic host libraries to virtualize the topology of the NAND array. The firmware also transparently handles and abstracts various NAND properties and programming requirements, including geometry and vendor-specific attributes. This functionality enables Forward Compatibility to support different NAND devices.

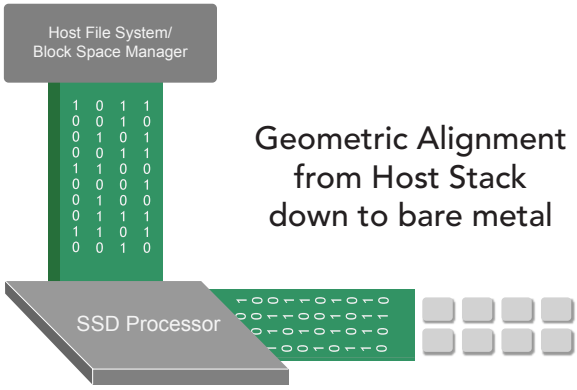


Symphonic eliminates latency spikes in the most challenging workloads (mixing writes with random reads), achieving a magnitude improvement in QoS

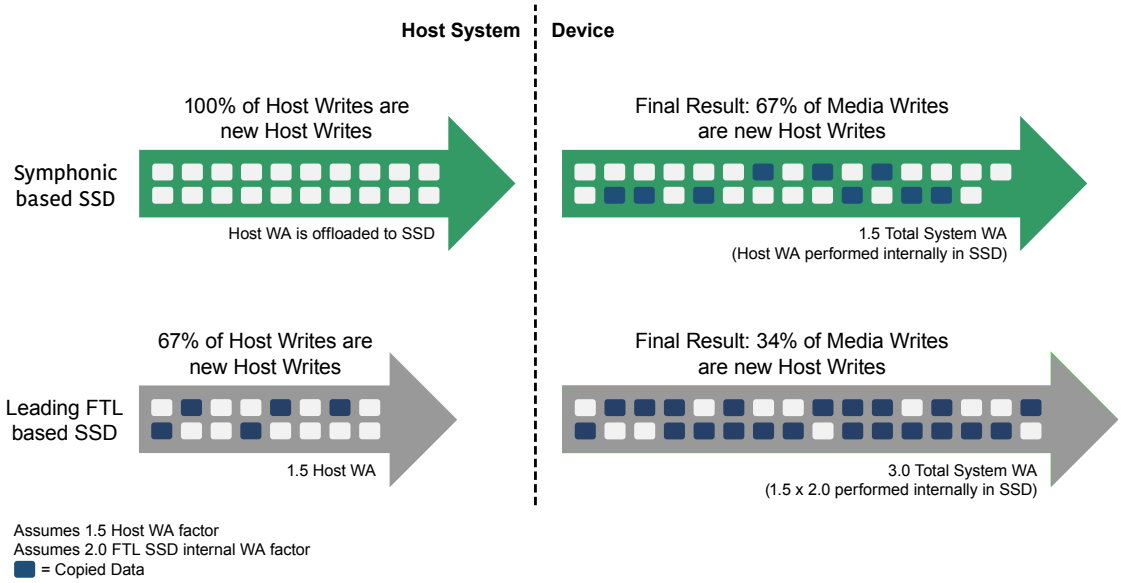


Symphonic achieves >80% improvement in bandwidth and IOPS over leading FTL SSDs

By taking a system-level approach to software architecture, Symphonic utilizes SSDs to provide the performance, latency, and endurance that can otherwise only be achieved with fully custom hardware platforms. Symphonic's coherency and geometric alignment enables massive parallelization that avoids collisions and unpredictable latency spikes.



Geometric Alignment from Host Stack down to bare metal



Reduces Write Amplification factor by >75%

Because Symphonic operates in host address space and performs garbage collection processes coherently under host control, only one level of write amplification occurs, instead of the double write amplification penalty incurred with FTL SSDs.

And because Symphonic firmware turns the SSD into an offload engine, the host's write amplification is transferred onto the SSD.

Consequently, 100% of host writes to the Symphonic SSD are for new data, eliminating costly copy/move overhead between the host and SSD.

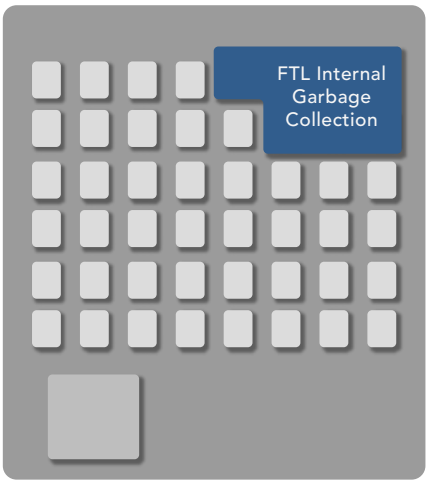
The final result is not just dramatically higher performance, but fewer writes to the Flash media which minimizes wear out and extends usable product life.

Less overprovisioning reduces acquisition costs

Enterprise class FTL SSDs typically reserve 15% of the SSD's raw Flash just for performing internal garbage collection operations. This capacity allocation is required to reduce unpredictable latency spikes (QoS), wear out, and improve IOPS/bandwidth.

Symphonic does not require overprovisoning for internal garbage collection, and consuming the typical associated 15% of reserve capacity, but provides dramatic improvements in each of the same metrics this overprovisoning attempts to address:

- Magnitude improvement in unpredictable latency spikes
- Typically reduces write amplification (wear out) factor by 75%
- More than 80% improvement in IOPS/Bandwidth



Radian Memory Systems, Inc.
5010 North Parkway Calabasas, Unit 205, Calabasas, CA 91302
Tel 818-222-4080 Fax 818-222-4081
sales@radianmemory.com www.radianmemory.com