



Flash Memory Summit

SARC-102-1: Application Acceleration Methodologies to Enterprise Storage Architectures

Brent Yardley

IBM STSM, Storage Hardware Architect, Master Inventor

yardleyb@us.ibm.com



Brent Yardley is a Senior Technical Staff Member and Master Inventor with IBM System Storage, where he focuses on developing All Flash Arrays (AFAs). Brent is currently the overall Chief Hardware Architect responsible for the hardware architecture, design, and integration of IBM's storage hardware products and planning future generation storage platforms. He specializes in system designs that integrate multiple I/O protocols, FPGAs, and ASICs, and is an expert in both hardware and software design and system integration.

A 22-year veteran of IBM, Brent holds multiple patents focused on storage architectures and solutions. He has an extensive background and understanding of the architecture, design, and implementation of highly available storage-based systems and solutions. He has earned BS degrees in both Software and Hardware Engineering from the Oregon Institute of Technology.



General Architectural Design

- High Availability with no single point of failure
- Every component must be concurrently maintainable
 - There are some exceptions like midplanes
- All repair events can occur while access to data is maintained
- All data must be preserved on power loss
- Density of design must be preserved regardless of physical enclosure size. These include design points such as:
 - 12 U.2 drives / U
 - 4 HBA slots / U
 - ~1kW / U
 - 1m maximum depth, including cable management

- Early architectures for high performance system, the IO data path was completely in hardware, ASICs, FPGAs, etc.
- In most modern designs, a general-purpose CPU architecture has been used to allow for additional computation of the data
- In future designs, the CPU involvement will again be reduced and possibly eliminated, as additional hardware offloads are added
- What is old is new again!



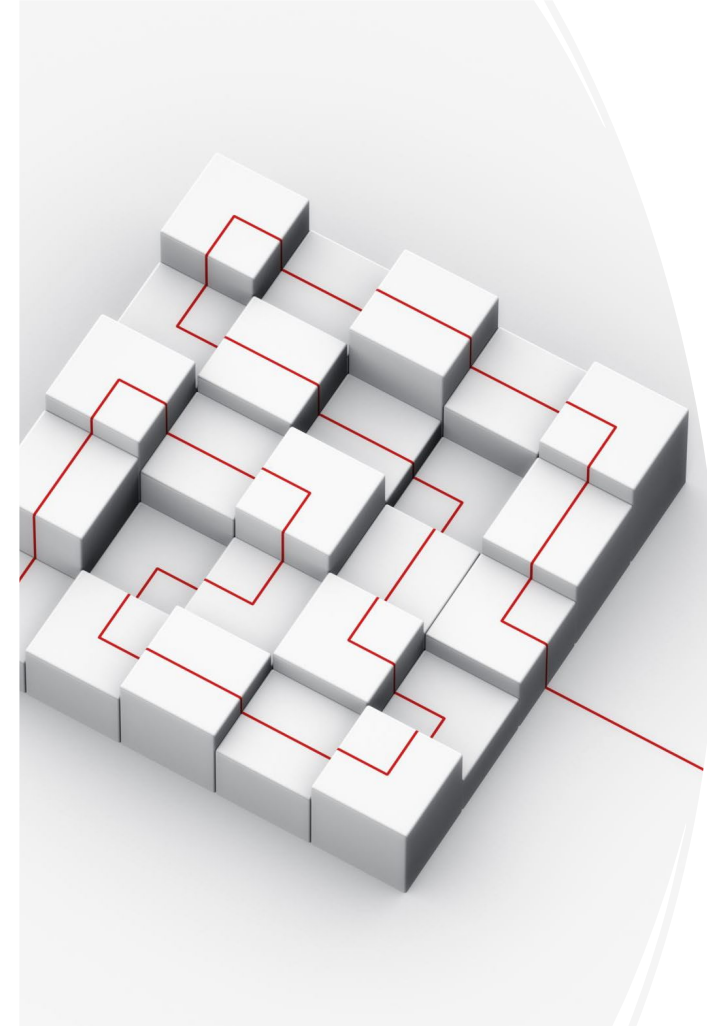


Power loss, the struggle to preserve data...

- With write cache, and the ability to early complete write transactions, the ability to preserve data on power loss can be solved in several ways
- At the enclosure level, storage designs can use a battery architecture with a destage in place design
 - Other enclosure level options can use NV-DIMMs
- At the drive level, NV-DIMM type architectures, where write data is flushed to NAND is can be used, with the assistance of capacitive energy devices,
 - persistent memory is another option and can be done used with significantly less power requirements, eg. MRAM.
- However, as memory systems increase, and write cache increases, there is a need to change the thoughts of how this is managed
 - CXL with persistent memory devices may be a good answer

Why One Enclosure Size Doesn't Fit All?

- Initially a single 2U enclosure was designed to fit all NVMe use cases across market segment
 - This allowed for significant benefits on development costs and reuse of design, however
 - Confusion between the client value across the segments
 - They are all are 2U, they all have the same number of drives, same memory configurations, what's the difference?
- Conclusion was to differentiate the segments by
 - Hardware capabilities (memory, CPU, PCIe)
 - Physical Form Factor, (1U, 2U, 4U)
- Provide same per U density for HBAs and drives



IBM FlashSystem NVMe[®] Family 2022

- Entry



FlashSystem 5200

- Mid-range



FlashSystem 7300

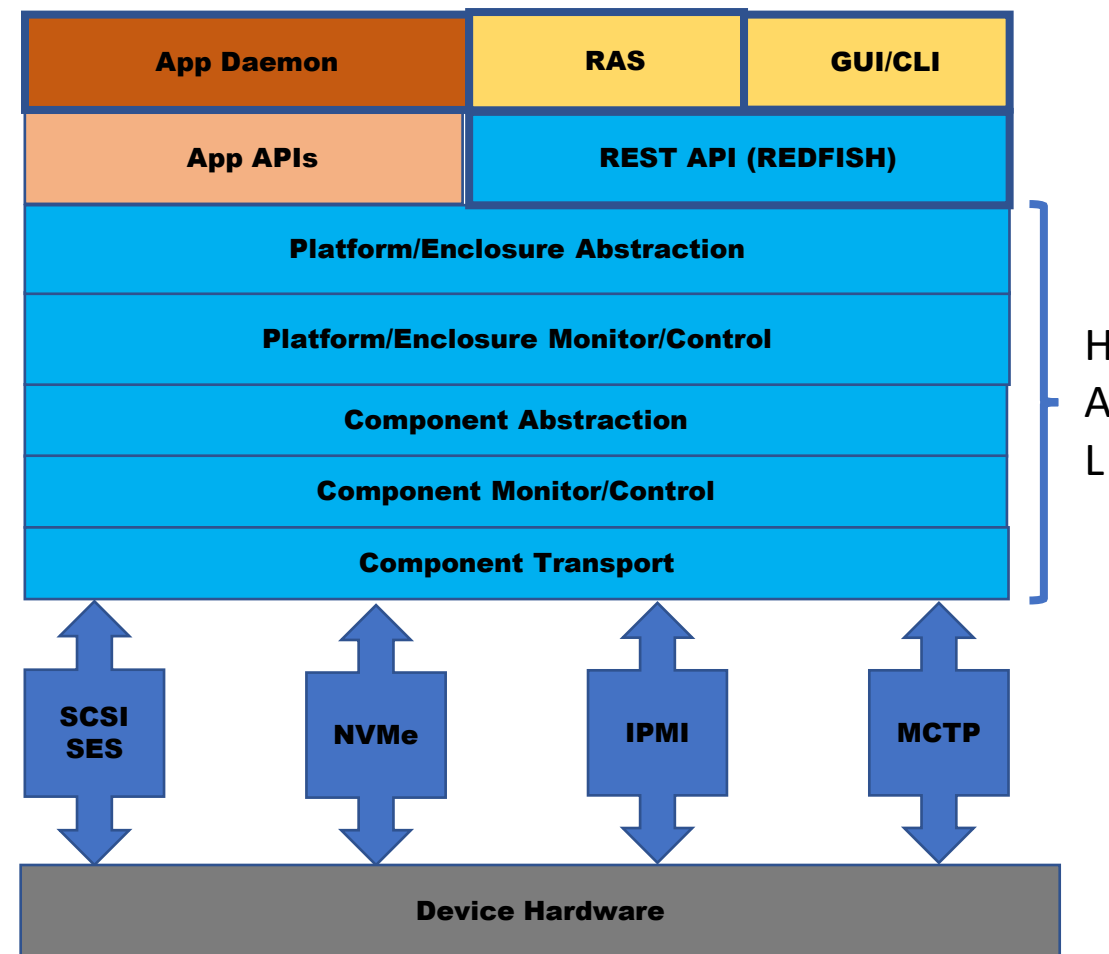
- High-end



FlashSystem 9500

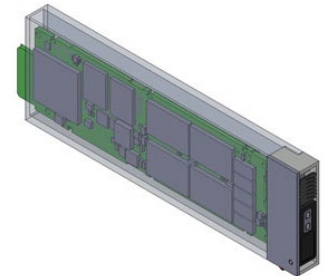
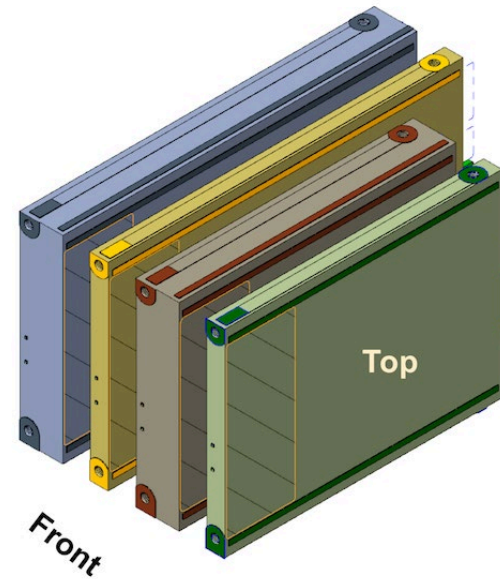
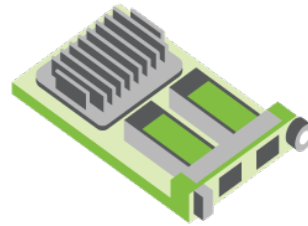
System Management Stack

- **App Daemons**
 - Sends SCSI or NVMe device protocol specific APIs.
- **App Application Program Interface**
 - Creates a set of device protocol agnostic asynchronous REST APIs.
- **Platform/Enclosure Abstraction**
 - Calls to this layer are thru an asynchronous REST API
 - Monitoring component statuses does not impact hardware
 - Provides RAS, UX, and App specific APIs
- **Platform/Enclosure Monitor/Control**
 - Processes component controls and status changes
 - Keeps component statuses current to events
- **Component Abstraction**
 - Provides generic interface into all components (capabilities, attributes, features)
- **Component Monitor/Control**
 - Device specific monitor and control
 - Can run older tsXXX commands
- **Component Transport**
 - Formats requests/replies into supported protocol
 - Handles error conditions and retries
 - Provides unique transport to hardware specific interfaces



And What's Next?

- Maintain differentiation between each segments
- Redesign the architectures to support emerging form factors
 - U.2 → E3
 - HHHL → OCP NIC 3
 - M.2 → E1.S for Boot
 - E3 2T Accelerators
- CXL Enablement
- NVMe HDDs?



When you interact with IBM, this serves as your authorization to Flash Memory Summit or its vendor to provide your contact information to IBM in order for IBM to follow up on your interaction.

IBM's use of your contact information is governed by the IBM Privacy Policy.

