

# SSD Firmware Complexities and Benefits from NVMe

Steven Shrader





# Agenda

- Introduction
- NVMe architectural issues from NVMe functions
- Structures to model the problem
- Methods (metadata attributes) to make the structure work well
- Attributes for a full system design
- Conclusion



# Introduction

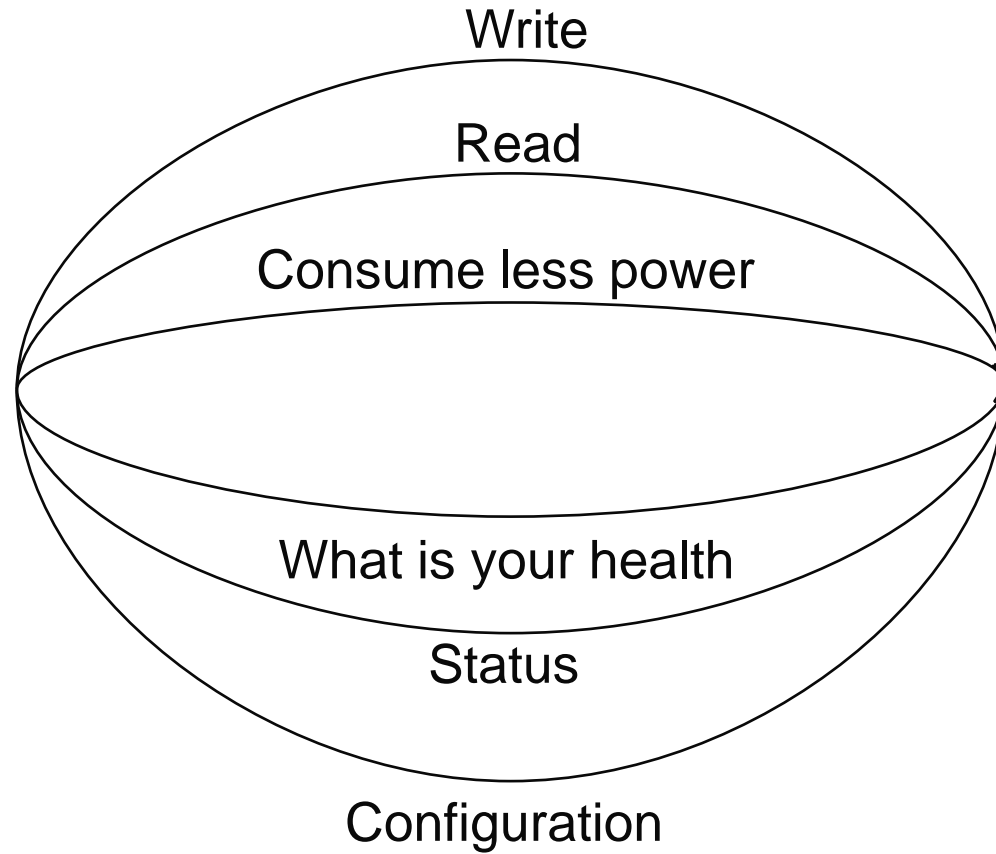
- NVMe is the solution to the interface performance bottleneck for high performance SSD's.
- NVMe through PCIe allows performance scalability in latency and bandwidth. (1M IOPS)
- Adds new functionality for virtualization and power control.



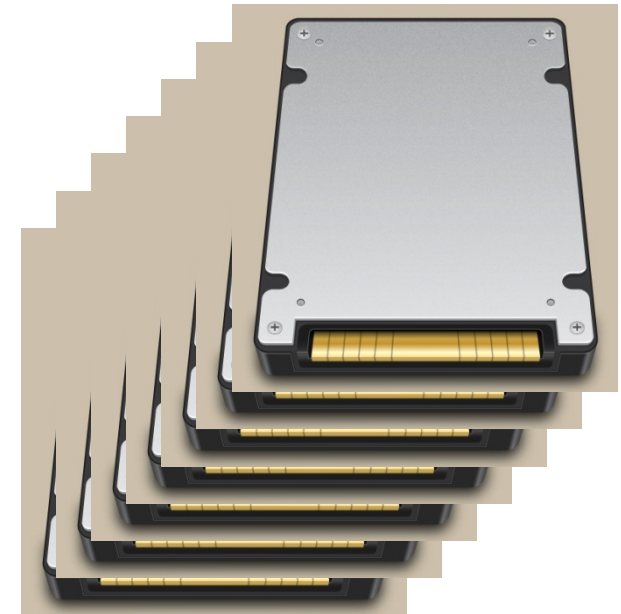
# NVMe architectural issues from NVMe functions



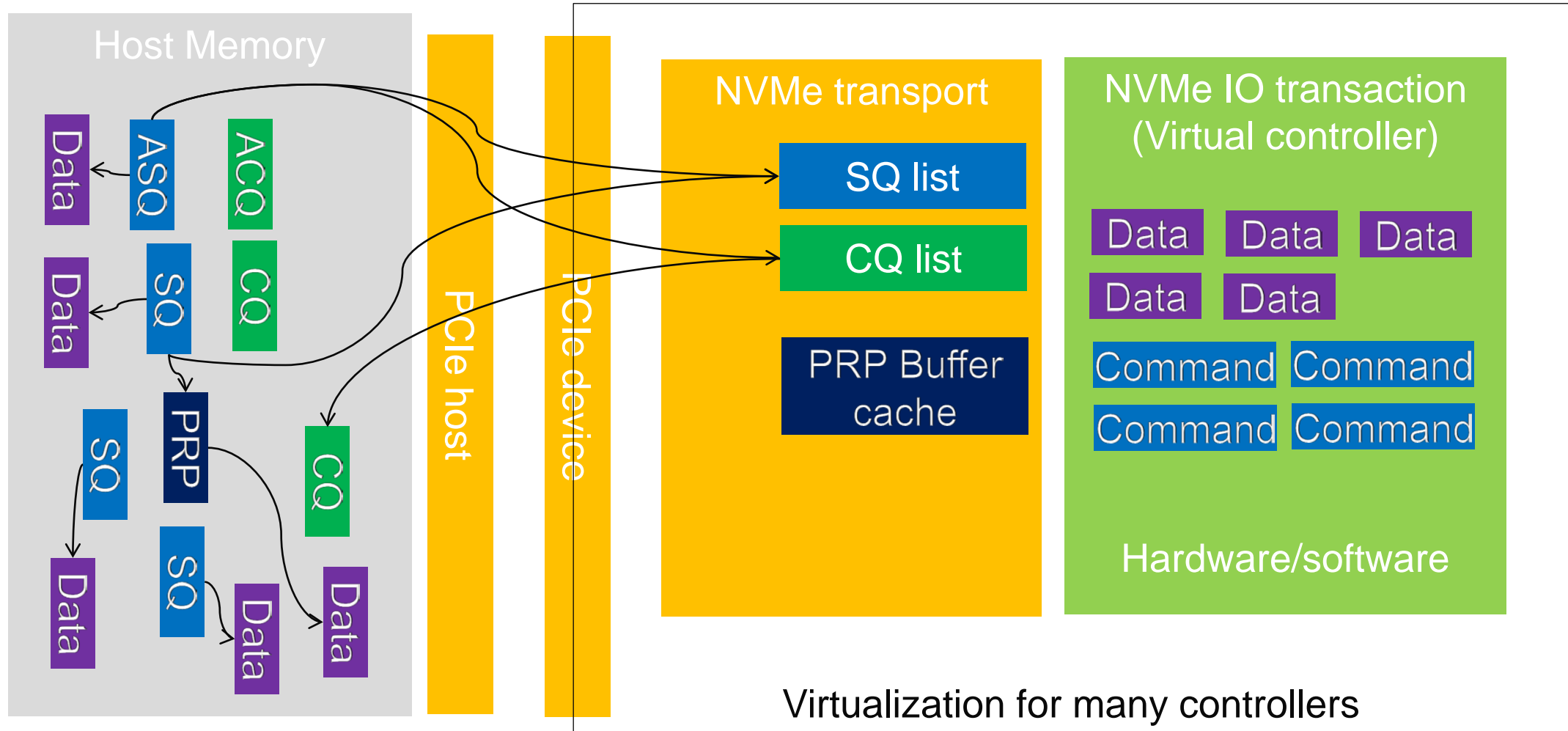
# Aspects of NVMe functions



Many virtual drives through one interface



# NVMe architectural issues from NVMe functions



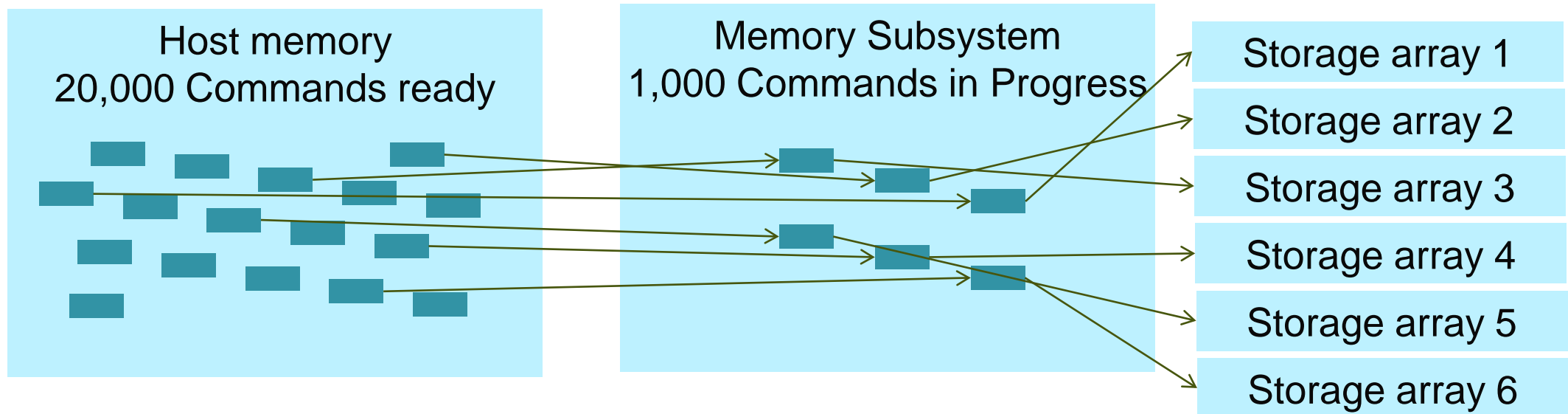


# Structures to model the problem



# Structures to model the problem 1

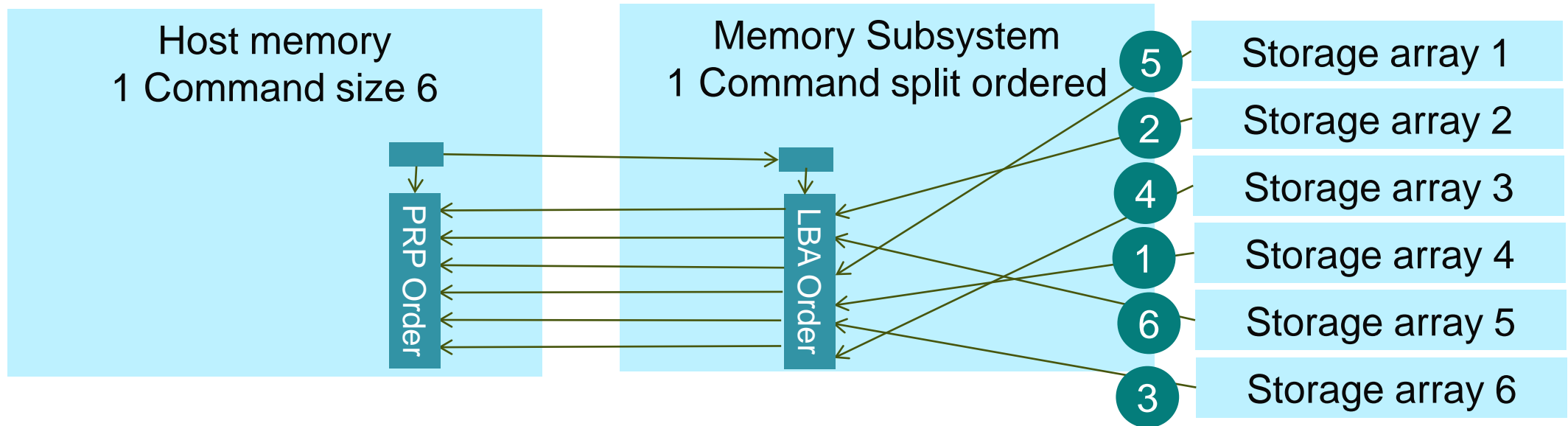
- Limit the total number of queues and entries in the queue by the requirements of the host system.
- Limit the total complexity of the interface by specifying the max number of parallel commands.



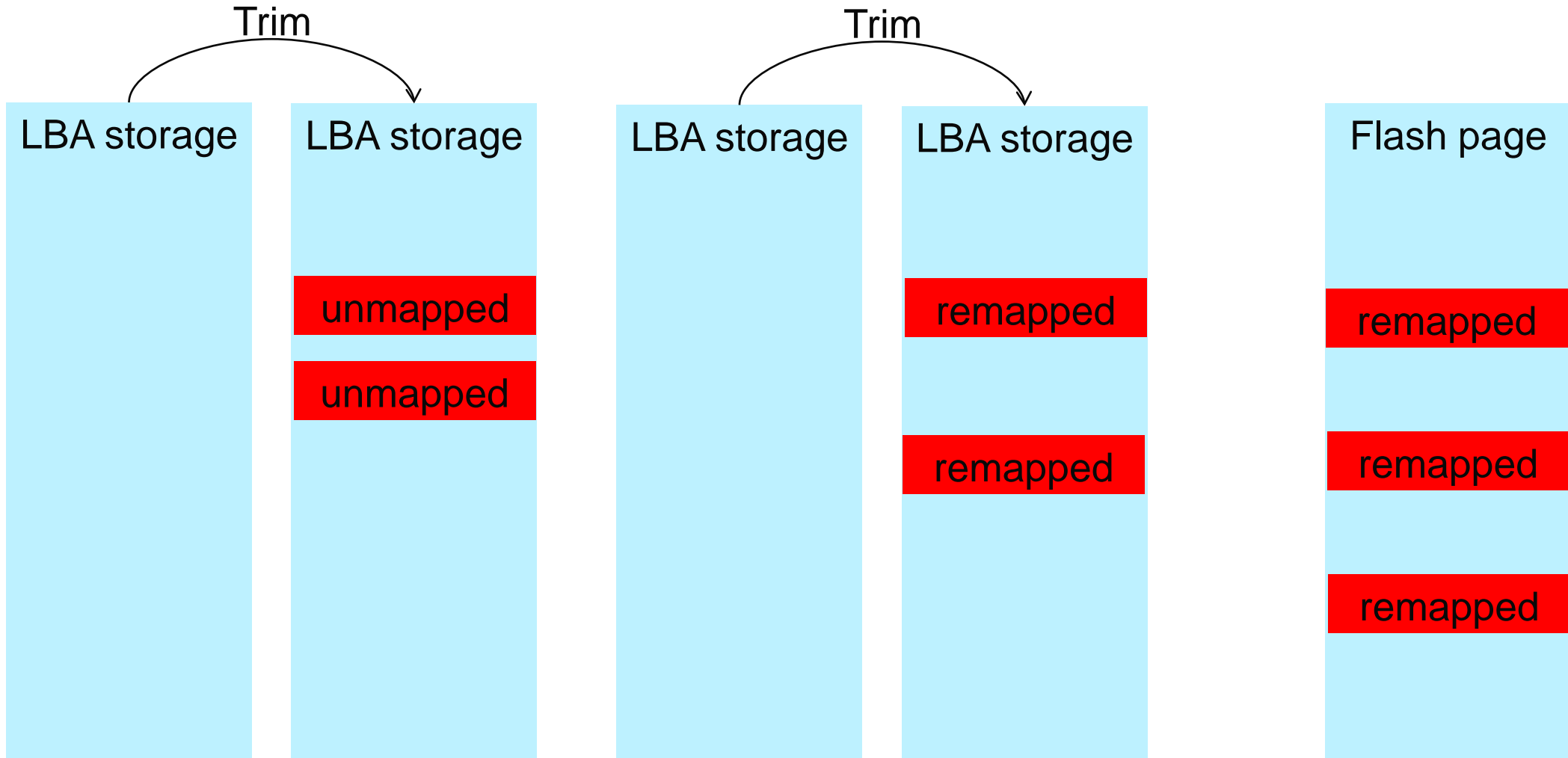


# Structures to model the problem 2

- Allow the controller to transfer the command data in managed units in random order to take advantage of cache or retrieval order from flash memory.
- Need the ability to time schedule buffer and or buss usage to achieve max throughput.



# Trim

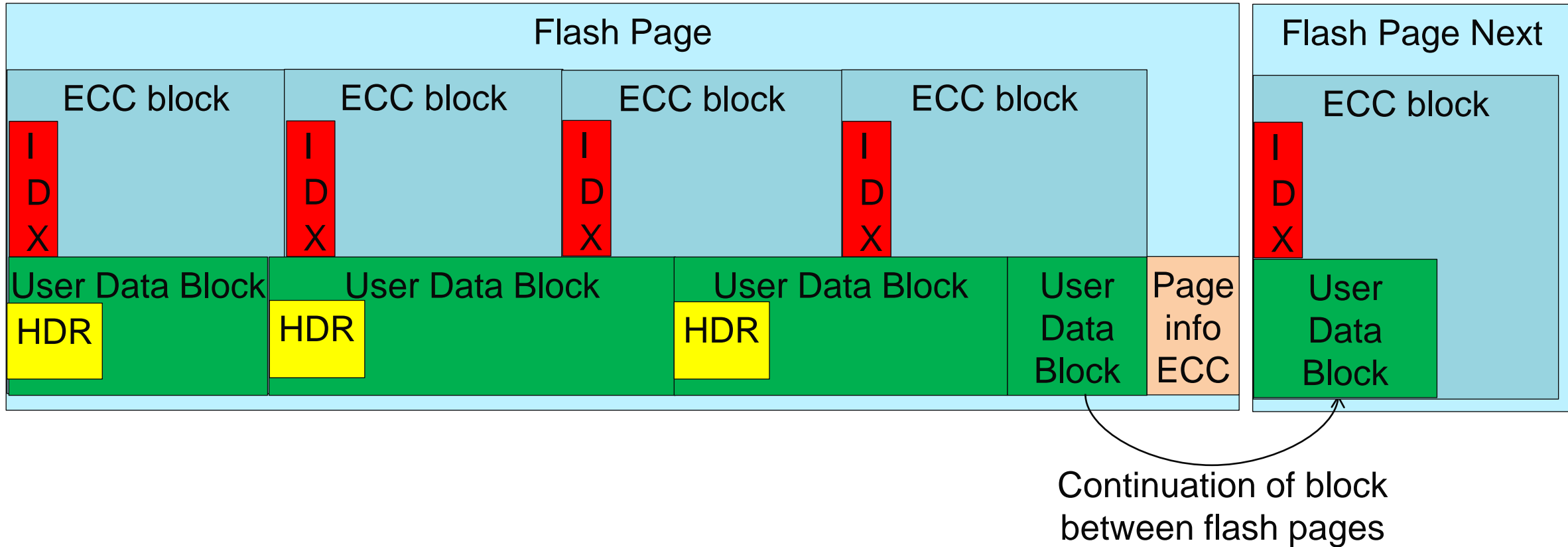




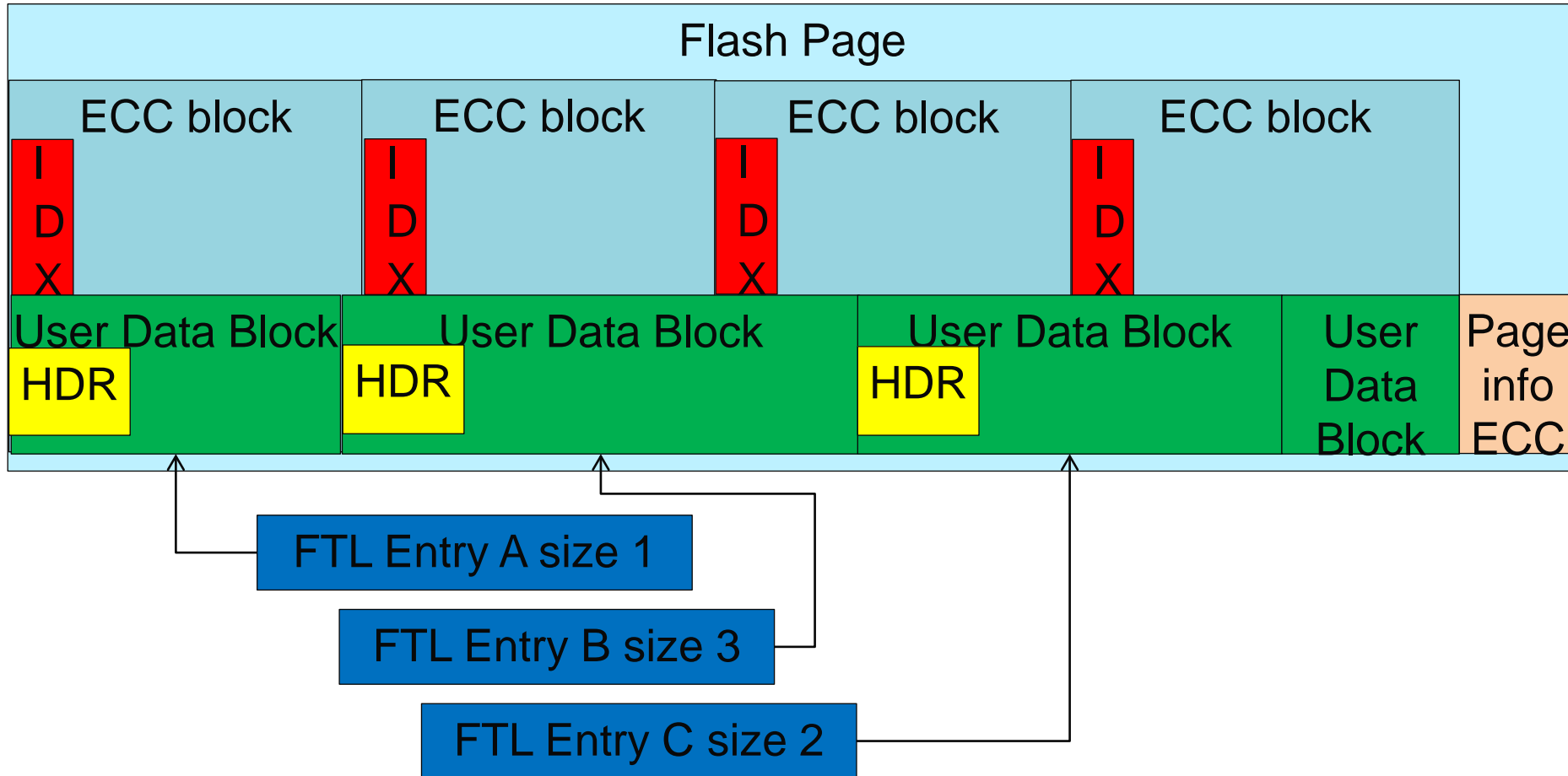
**Methods (metadata attributes) to  
make the structure work well**



# Methods (metadata attributes) to make the structure work well 1

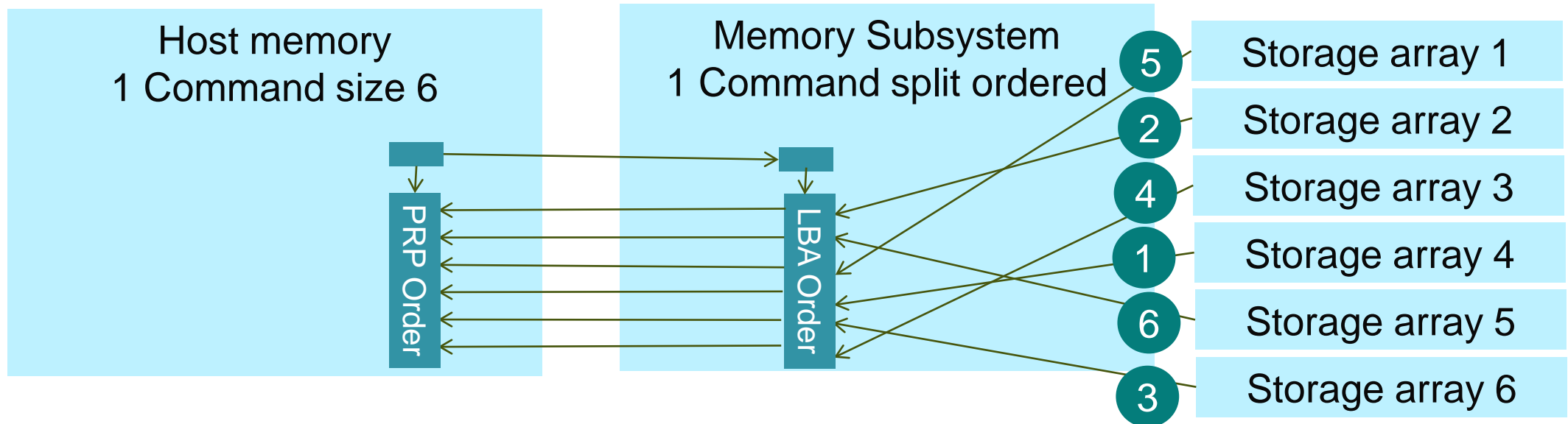


# Methods (metadata attributes) to make the structure work well 2



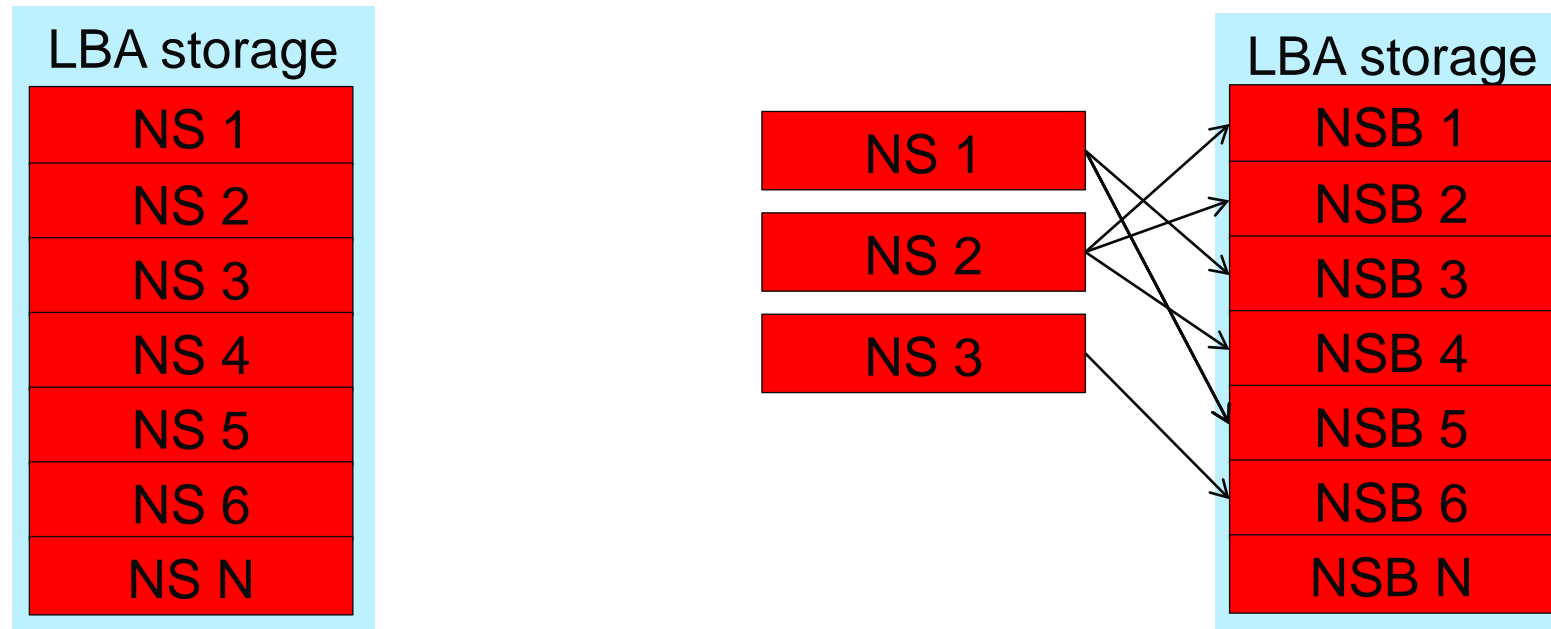
# Methods (metadata attributes) to make the structure work well 3

- Command expansion is the process of relating LBA's to physical storage.
  - This is done in “Atomic write units” for each command.



# Methods (metadata attributes) to make the structure work well 4

- Namespaces can be any block of storage composed of a type of memory.
  - Most generally, drive composed of a single storage medium, this will be another layer of mapping in front of the physical memory. Either as a set of blocks or as a linear list of physical blocks





# Attributes for a full system design





# Attributes for a full system design 1

- Full system must include methods that work well from a system perspective
- Use full bandwidth of the PCIe by structuring the hardware to do data transfer to and from the host at the same time in a pipeline method
- For enterprise, a consistent performance for read/write from a host perspective is a big desire, therefore we need background processing to execute in its own timing to not interfere with host transactions read/writes
- Best if all storage can handle any attribute of a namespace
- Best if you can structure storage into as many namespaces as possible

# Attributes for a full system design 2

- Best if you can respond to low power requests from host system to achieve the best performance on the available power
- Include methods to analyze and understand the performance, power, errors, functions, actions of an implementation at speed



# Conclusion



# Conclusion

- We need to structure all hardware/software/interfaces to allow for all attributes in this presentation
- We need to combine high performance with configurability to achieve the highest performance based upon the available resources for a specific implementation
- Debug and analyze all components of the system

cadence your complete NVMe solution



**cādence<sup>®</sup>**