SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2014

Deduplication, Compression and Pattern-Based Testing for All Flash Storage Arrays

Peter Murray - Load DynamiX

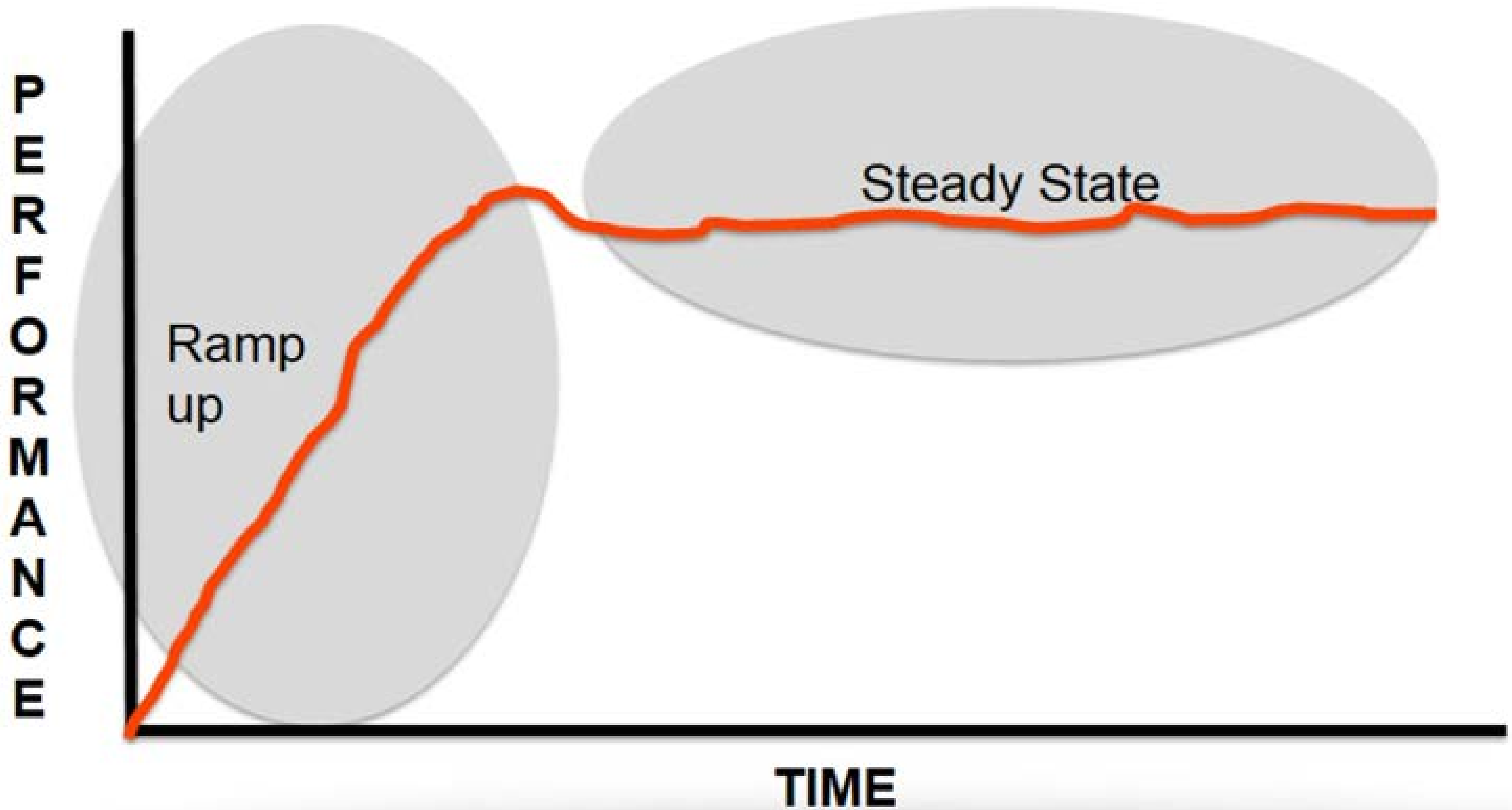
Leah Schoeb - Evaluator Group

Introduction

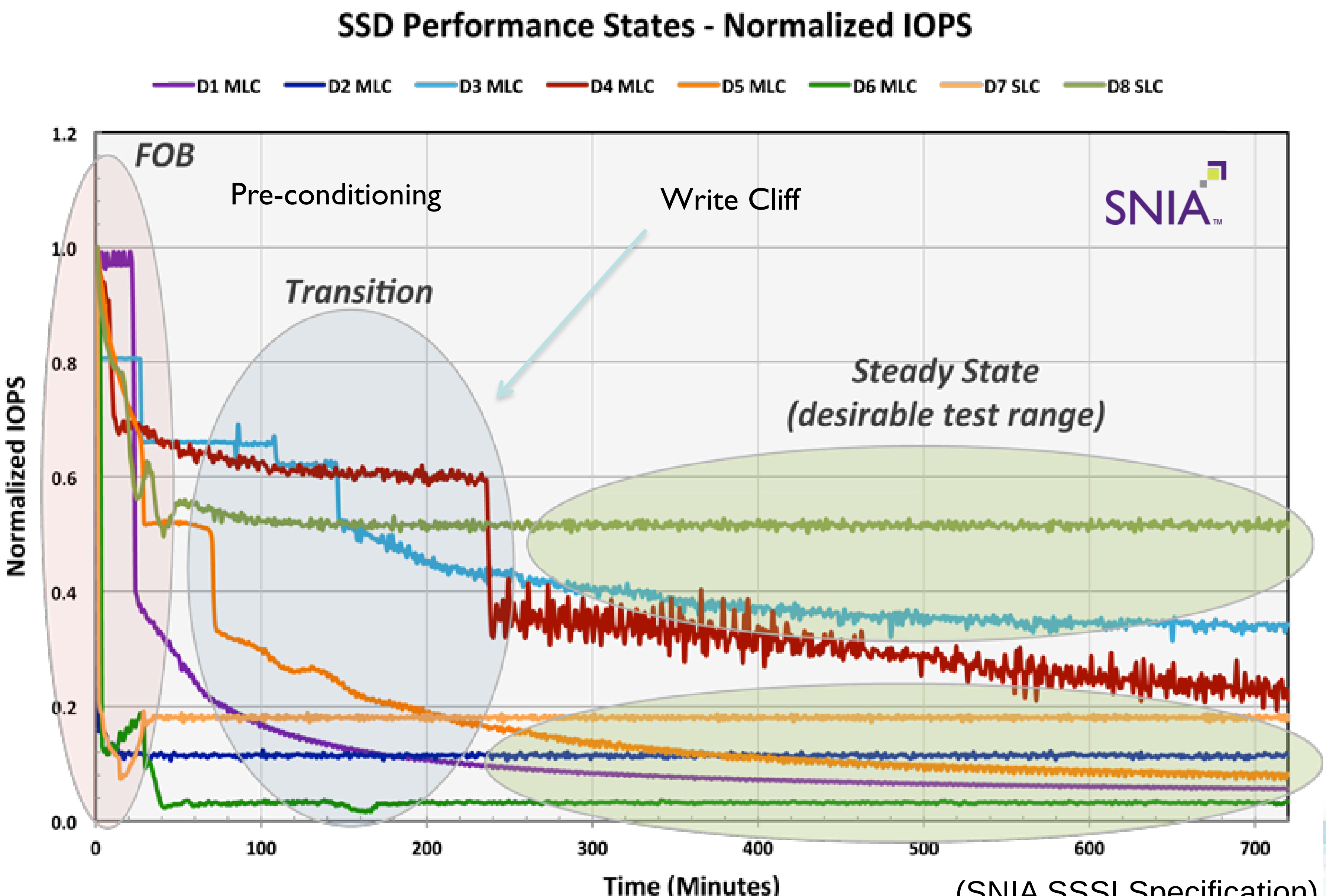
Advanced AFAs are a Different Animal

- Flash behavior is unique
- AFAs have a different performance curve
- Advanced AFAs do not merely store data
  - Most perform extensive metadata processing
    - Deduplication
    - Compression
    - Elimination of repeating character strings
- These new arrays require a new performance testing methodology

Traditional Disk Performance Curve



Flash Performance Variations

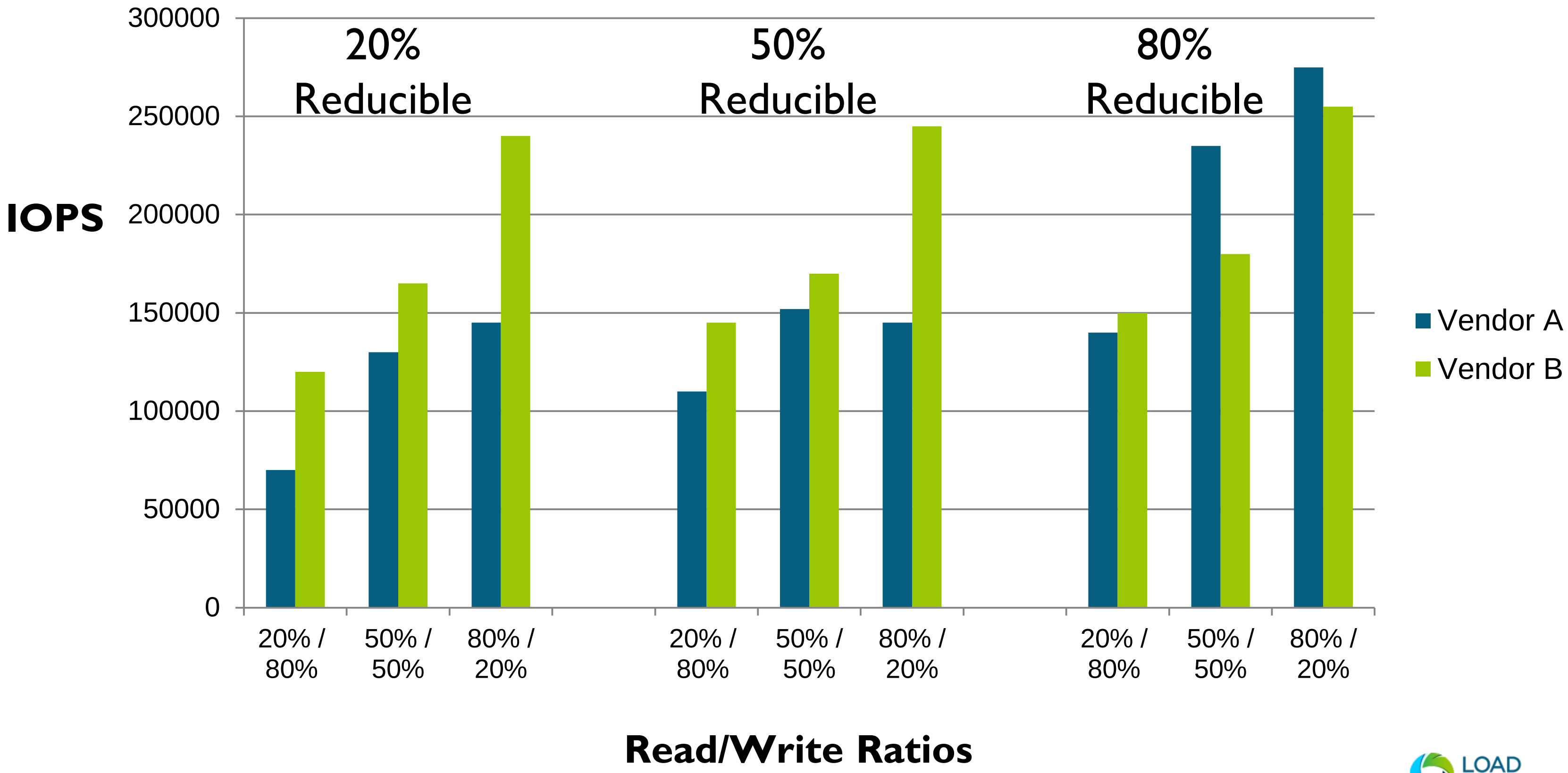


(SNIA SSSI Specification)

Methodology In Action

IOPS Comparison for 3 Groups of Data Patterns & R/W Ratios

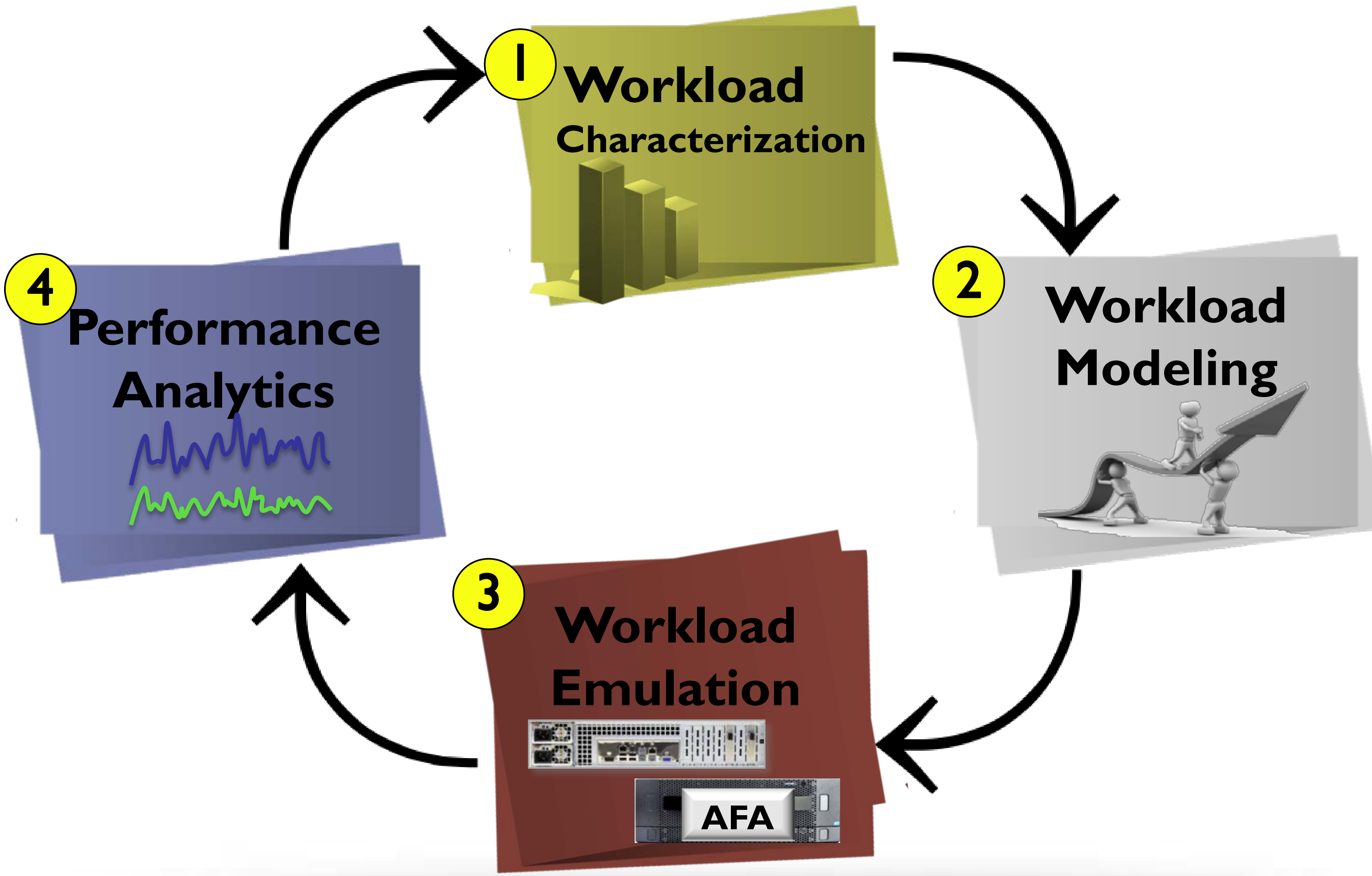
Which is best?
Depends on your workload.



Implementing a Methodology to Achieve Realistic Workload Emulations

- Methodology is a means to an end
  - Effective application workload modeling
  - Benchmarks
- Validation takes SSS TWG methodology to a new level
  - Testing that emulates application workloads
  - Workload combinations that emulate the I/O blender
  - Requires complex testing capabilities
  - Requires correlated results

New Approach to Validating AFAs



Primary Methodology Elements For Testing an AFA

Deduplication

- Approaches vary by manufacturer
- Dedupe block size
  - Larger block size speeds processing
  - Smaller size can dedupe better, but requires more processing
- Ingest processing, post processing or both
- Deduplication in the presence of data skew

SDC 14

Compression

- Vendor implementations vary
  - Not as prevalent yet as deduplication
    - Increasingly being supported by vendors
- Performed during ingest
- Compression block sizes may increase overall compressibility
  - Vendor dependent

Eliminating Repeating Character Strings

- Repeating characters stored as metadata
  - Metadata identifies:
    - Character
    - Number of repetitions
- Performed during ingest

SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2014

Methodology Overview

Methodology Elements

- Pre-conditioning
- Creating a realistic data set
- Writing to create an application data set on array
- Writing to exercise the array emulating an appropriate workload
- Other tests to emulate realistic, simultaneous writing and reading

Pre-Conditioning

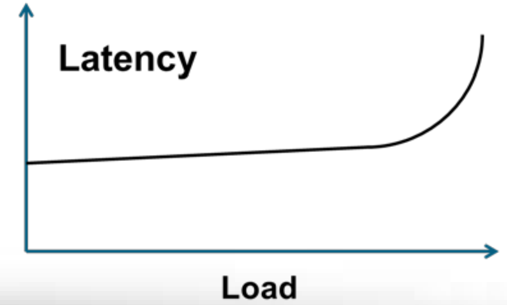
- Involves breaking in entire flash array
  - Writing to every cell to achieve steady state
  - Helps to ensure garbage collection during main test cycles
- Goal: create a realistic data set
  - Dedupeable and non-dedupeable blocks
  - Compressible and non-compressible blocks
  - Combined using varying block sizes
  - Written to emulate hot spots and drift
  - Written with appropriate dedupe/compression ratios

Write Performance Tests

- Exercising array like an application does
  - Writing at high load to find limits
  - Writing using a data stream relevant to the data set
  - Writing to emulate long-term application access
- Goal: Exercising the array realistically
  - Using a variation of the pre-conditioning data set
  - Writing with same levels of data reduction
  - Using multiple block sizes
  - Including hot spots and drift to emulate temporality

Read/Write Workload Tests Scenarios

- Tests that write and read simultaneously
  - All-write tests do not exercise an array the way an operating application does
  - Reading must be combined with writing for realism
    - Tests using all-write data patterns, but reading also
  - Run at expected application load
- "What if" testing to determine performance limits
  - Magnifying the load to test future expected loads



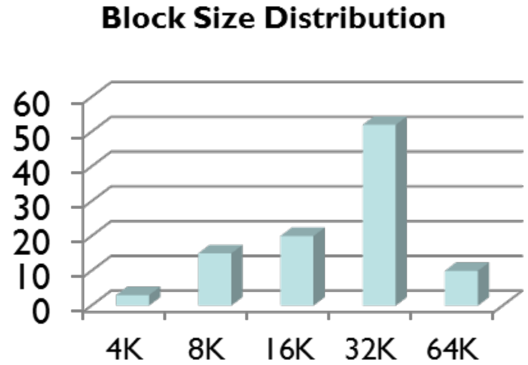
Methodology Components

Block Size

- Block sizes vary by application and operation
  - 25K-40K average block size is common
    - But, no application uses uniform block sizes
  - Sizes vary according to operations
- OLTP transactions typically small
- Analytics, reporting typically larger

Block Size (continued)

- AFA methodology should reflect real access
  - Single application
  - I/O Blender (multiple, usually virtualized, applications)
  - Either model requires multiple block sizes
- Should reflect application/blender access distribution
  - E.g. 3% 4K, 15% 8K, 20% 16K, 52% 32K, 10% 64K



Hot Spots / Hot Bands and Drift

- Application access is not uniformly random
  - Hot spots are storage locations accessed more frequently than others
  - Hot spot regions drift over time
    - E.g. Index file growth as transactions are processed
- Hot Spot examples:
  - Index Files
  - Temp Files
  - Logs
  - Journals

Hot Spots/Bands and Drift (continued)

- Hot spot emulation example:
  - 1% of all access regions receive 35% of the IOs
  - 1.5% of all access regions receive 15% of the IOs
  - 2.5% of all access regions receive 15% of the IOs
  - 5% of all access regions receive 15% of the IOs
  - 7% of all access regions receive 10% of the IOs
  - 6% of all access regions receive 5% of the IOs
  - 7% of all access regions receive 3% of the IOs
  - 5% of all access regions receive 1% of the IOs
  - 65% of all access regions receive 1% of the IOs

Access Patterns

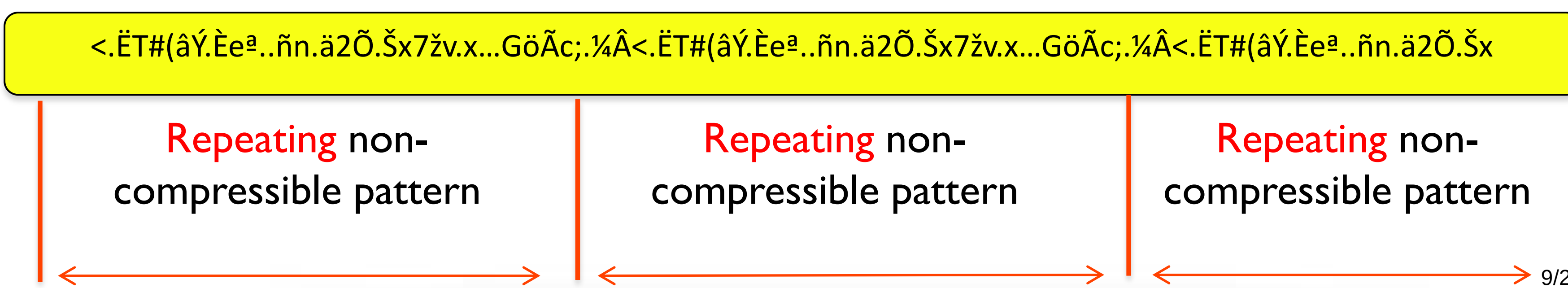
- Tests must reflect realistic access patterns
  - Should emulate real applications
  - Should avoid uniform random write distribution
  - Should use multiple block sizes
  - Should avoid unrealistic access patterns that skew towards systems that maintain larger amounts of reserve flash memory
- Should include testing in the presence of:
  - Backups
  - Snapshots
  - Replication

Complex Data Patterns

- Complex data patterns model workloads
- Pattern types:
  - Unique
  - Repeating
  - Uncompressible
  - Compressible
- Combined to represent data content representing:
  - Data set at rest after pre-conditioning
  - Data patterns that emulate traffic during operation

Data Content

- Data content patterns
  - Created before testing
- Data content streams
  - Written during testing
- Repeating and non-repeating patterns
  - Random
  - Compressible
- Varying pattern lengths



Thread Count and Queue Depth

- Both should increase during testing
- Should find max throughput for each:
  - Thread count (workers)
  - Queue depth (outstanding I/Os per worker)
- Should find max IOPs for each:
  - Thread count
  - Queue depth
  - Combination of threads and queue depth
- Should increase thread count/queue depths to find max array performance

New SNIA Technical Working Group

Solid State Storage System

Technical Working Group

(s4twg.snia.org)

(s4twg@snia.org)

Solid State Storage System (S4) TWG

- Address the unique performance behavior of Solid State Storage Systems (S4)
- Measure performance of inline-advanced features
- Measure performance of enterprise arrays vs. devices
- System wide housekeeping vs device level
- Caching and DRAM tiering

Charter

- Identify, develop, and coordinate standards to enable accurate performance measurement for solid state storage systems
- Produce a comprehensive set of specifications and drive consistency of measurement guidelines and messages related to solid state storage systems
- Document system-level requirements and share these with other performance standards organizations

Program of Work

- The TWG will develop a specification for measuring the performance of solid state systems.
- The TWG will develop a specification focused on solid state storage systems that support inline advanced storage features that directly impact performance and the long term behavior of the array.
- *Note:* This will build upon process methodology developed by the SSS TWG

Summary

Summary

- All-Flash Arrays are unlike disk-based arrays
- Data reduction dramatically changes performance characteristics
- Tests must include rich data content to be valid
- Tests must model real-world access patterns

Summary

- Tiered arrays are unlike all-flash arrays
  - This methodology valid for arrays that implement data reduction, but may not be appropriate for tiered arrays
  - A second methodology may be required, especially for tiered arrays that do implement data reduction
- Testing must be fair, unbiased and repeatable
  - “One size fits all” may not be fair to tiered arrays

References

- www.evaluatorgroup.com - “Measuring Performance of Solid State Arrays”
- www.loaddynamix.com – “Go Daddy White Paper: Storage Validation”