STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2015

# The Changing Storage Testing Landscape


Peter Murray

Load DynamiX

# Introduction

# Advanced AFAs are a Different Animal

- Flash behavior is unique
- Arrays that use flash memory have a different performance curve
- These new arrays require a new performance testing methodology
- So, what's different?

# What's different With an Advanced AFA?

- High speed
- High capacity
- Data reduction services
- Enterprise services
  - Clones
  - Snapshots
  - Replication

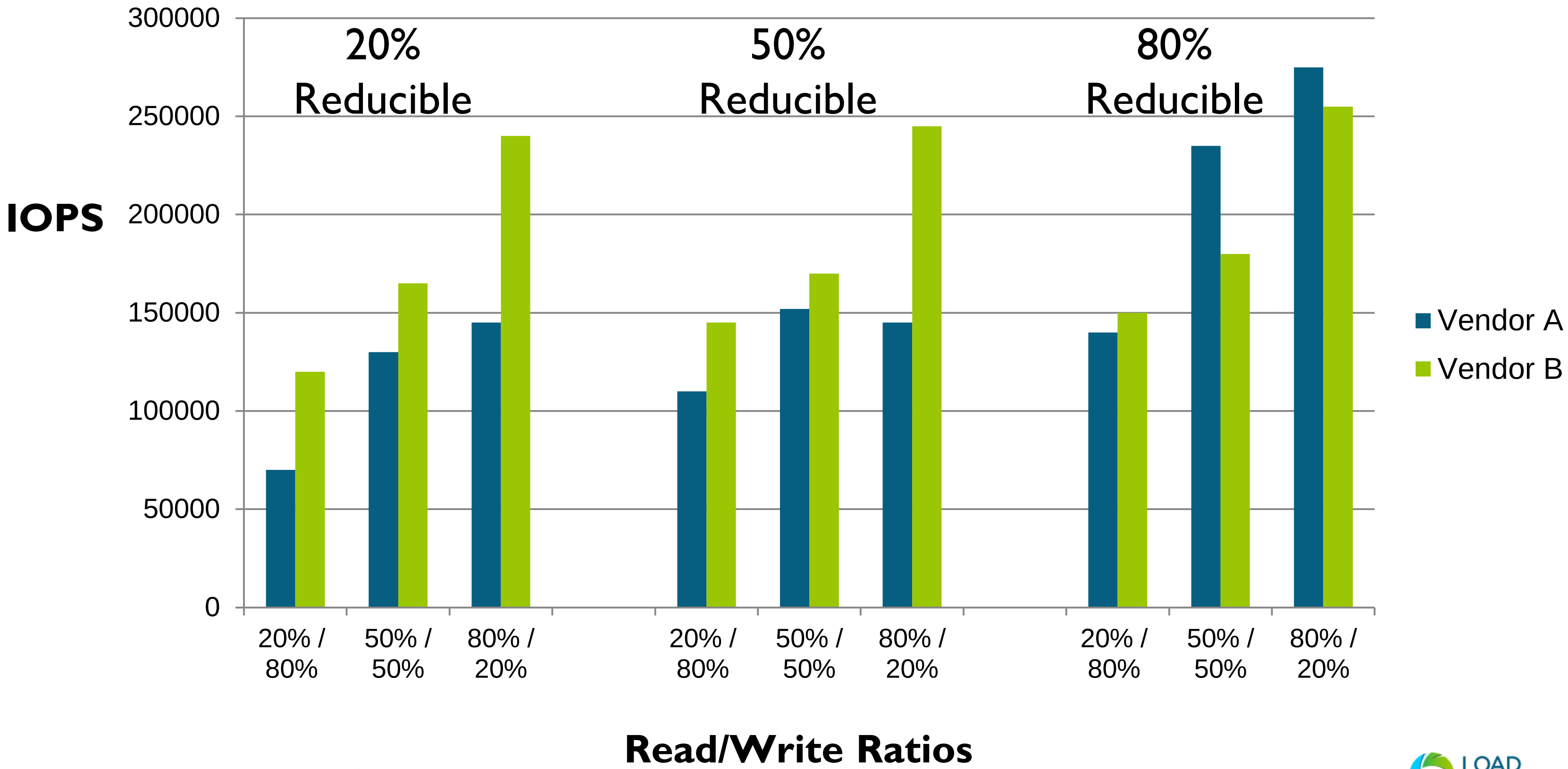
# This Methodology is for All Arrays

- All storage arrays can use this methodology
  - AFA, Hybrid or Magnetic Disk Arrays
- The goal is realistic testing
- Issues may arise with multiple apps on one array
  - Enough cache/SSD to maintain all hot spots?
  - Ability to move data to cache/SSD tier when needed?

# Array Performance Variations

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

**Which is best?**
**Depends on your workload.**



# Testing with Realistic Access Patterns

# Complex Access Patterns

- Performance varies with array types:
  - Magnetic-Media
  - Hybrid
  - All Flash
- An array should be tested with realistic access
  - Testing should emulate the application/s that will run on the array
  - Complex access patterns are required to create application test cases

# Realistic Access Patterns

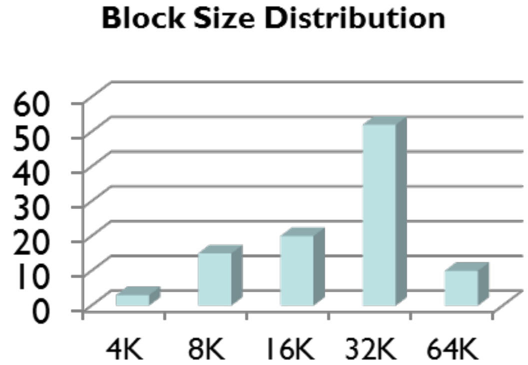
- Tests must reflect realistic access patterns
  - Should emulate real applications
  - Should avoid uniform random write distribution
  - Should use multiple block sizes
  - Should avoid unrealistic access patterns that skew towards systems that maintain larger amounts of reserve flash memory
  - Should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

# 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
  - Multiple virtualized applications (I/O Blender)
  - Either model requires multiple block sizes
- Should reflect real access pattern 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

# Thread Count, Queue Depth and Asynchronous I/O

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

# Testing with Realistic Data Patterns

# Complex Data Patterns

- Complex data patterns are also required to test realistically
  - Deduplication is widely implemented and must be accounted for
  - Compression is increasingly important also
- Data patterns are aggregated and sent to array
  - Patterns at rest emulate application patterns
  - Patterns in flight emulate application data flow



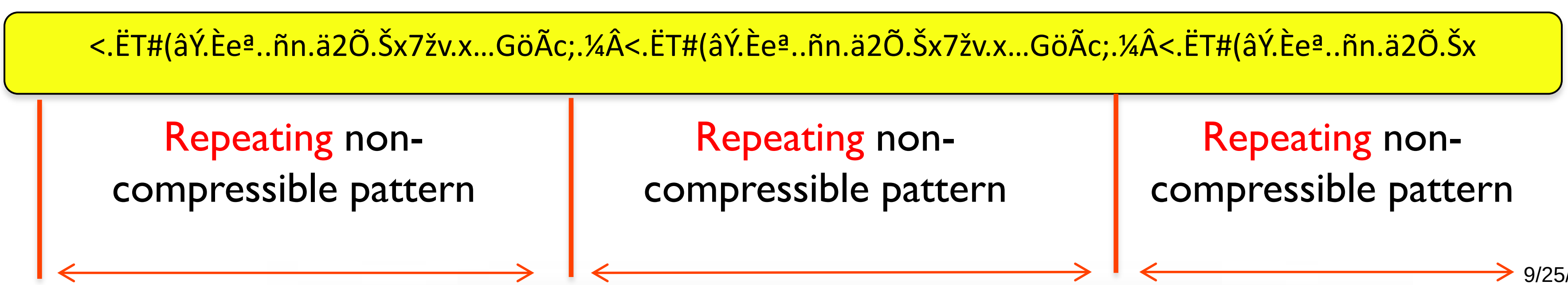
SDC 15

# 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



# 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

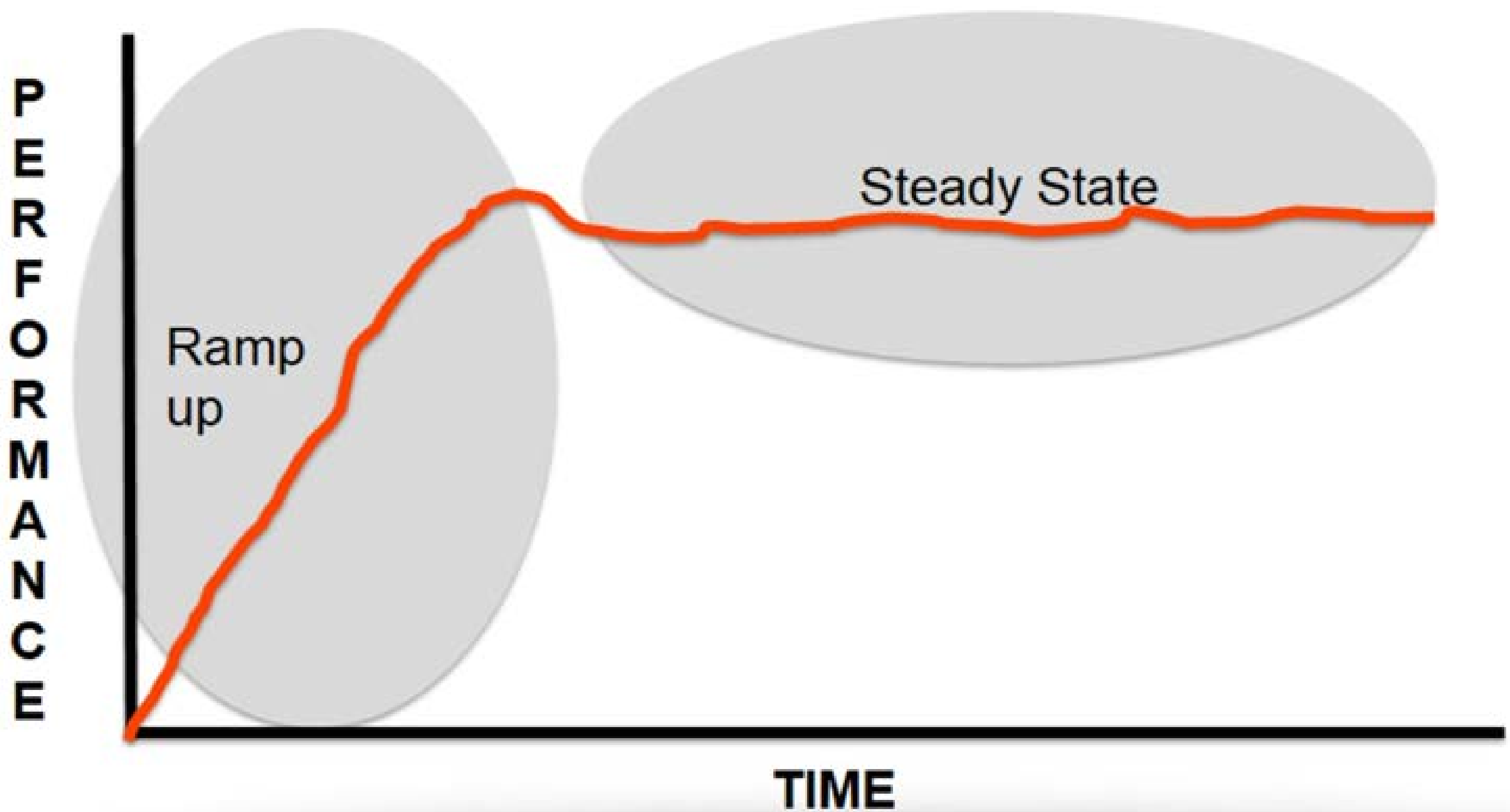
# Compression

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

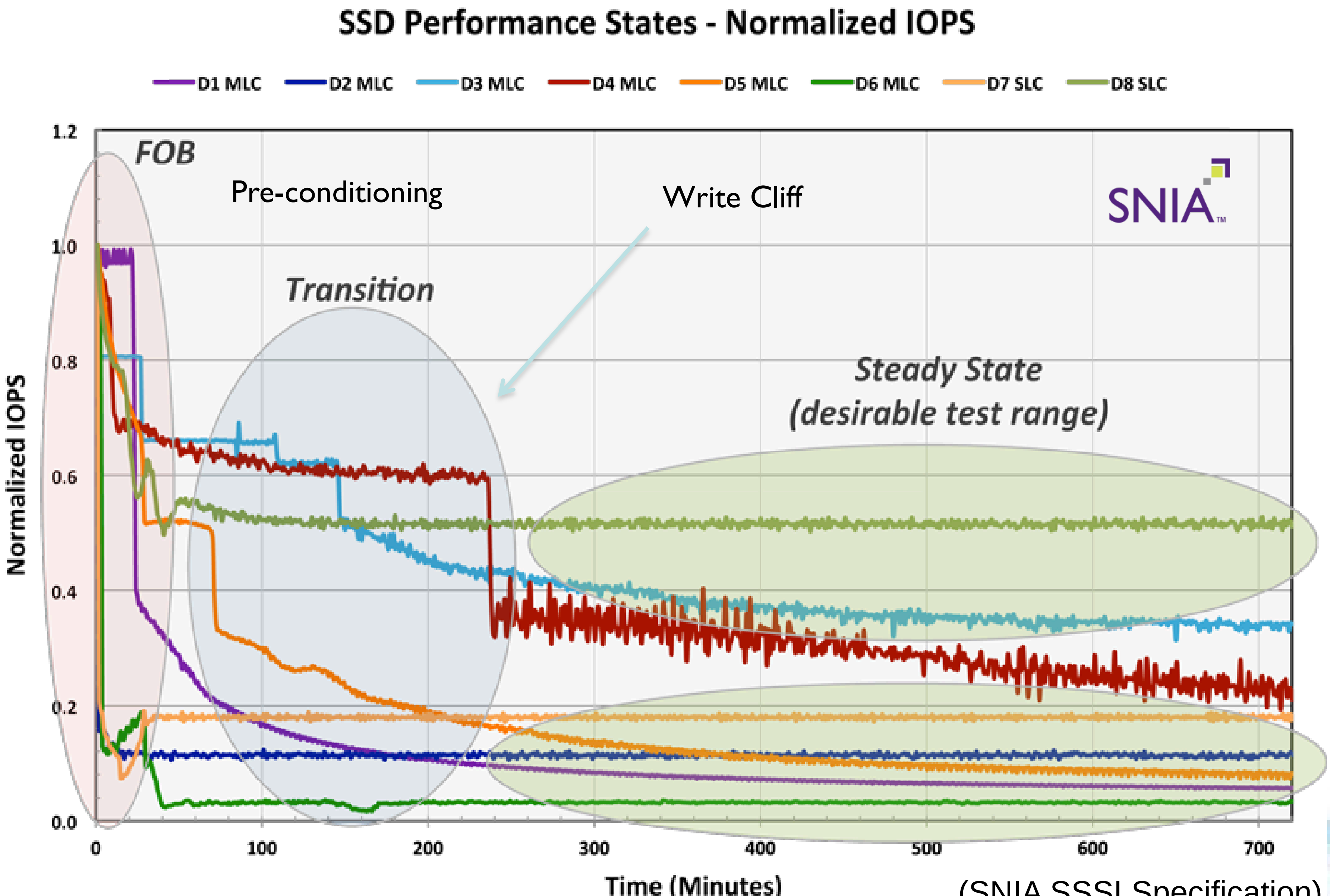
# Eliminating Repeating Character Strings

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

# Traditional Disk Performance Curve



# Flash Performance Variations

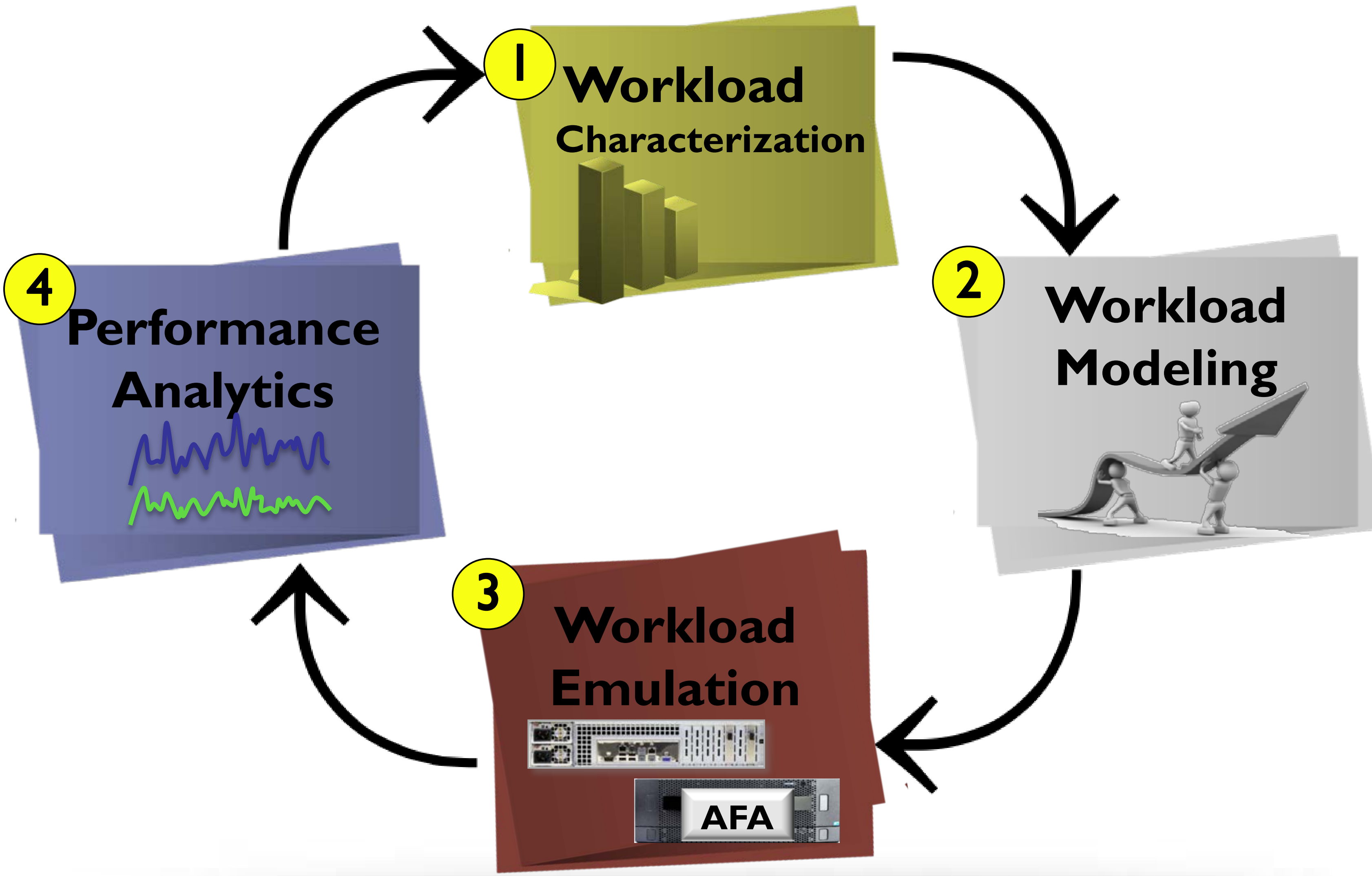



(SNIA SSSI Specification)

# 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

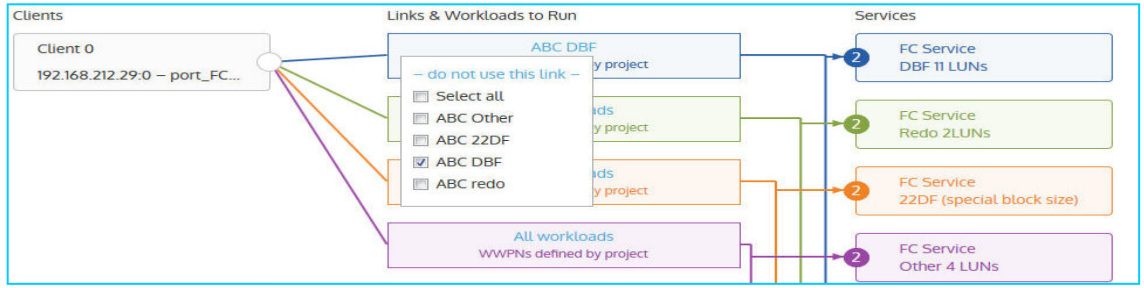



SDC 15

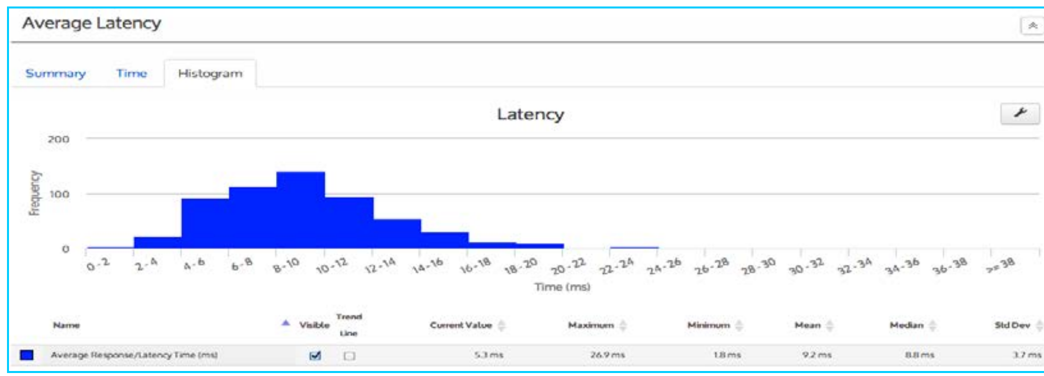
# Example Oracle Workload

# Sample Use Case - Characterization

1. Characterize composite workload from existing metrics … each line below represents a distinct workload, metrics captured from the existing (old) array

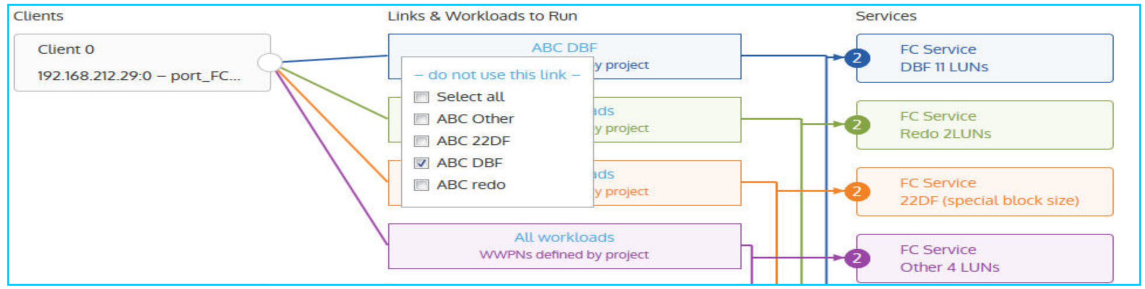



| Host IOs/sec | % Writes | %Reads | Avg I/O Size | Capacity (GB) |
|---|---|---|---|---|
| 522.1 | 0.4 | 99.6 | 19 | 256 |
| 448.5 | 0.1 | 99.9 | 16 | 256 |
| 316.6 | 5.2 | 94.8 | 19 | 256 |
| 297 | 0.9 | 99.1 | 29 | 100 |
| 235.6 | 4.8 | 95.2 | 17 | 256 |
| 220.2 | 5.6 | 94.4 | 20 | 256 |
| 201.4 | 1.5 | 98.5 | 237 | 256 |
| 165.7 | 5.1 | 94.9 | 19 | 200 |
| 91.9 | 6.2 | 93.8 | 17 | 100 |
| 90.3 | 27.7 | 72.3 | 48 | 200 |
| 70.2 | 96.9 | 3.1 | 28 | 32 |
| 68.1 | 99.6 | 0.4 | 14 | 32 |
| 7.6 | 17.8 | 82.2 | 105 | 256 |
| 6.3 | 88.5 | 11.5 | 13 | 10 |
| 2.8 | 98.4 | 1.6 | 347 | 256 |
| 1.5 | 17.7 | 82.3 | 2 | 33 |
| 0.1 | 11.5 | 88.5 | 4 | 33 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 2.5 | 97.5 | 0 | 33 |

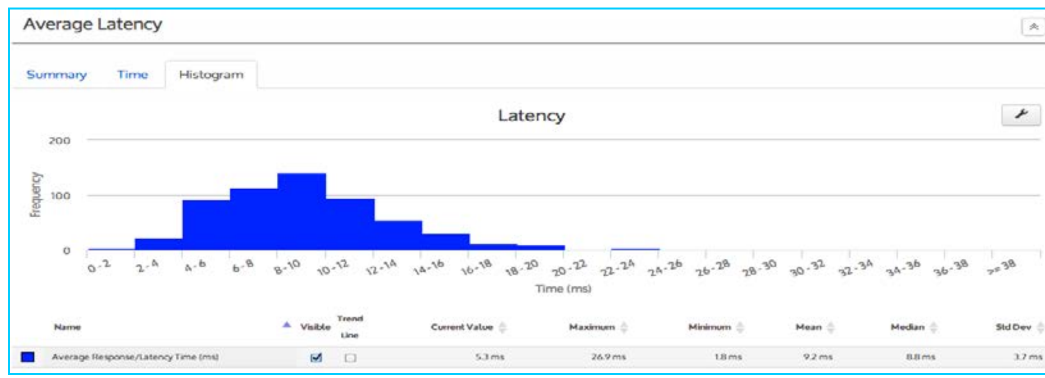


# Sample Use Case - Modeling

2. Model profile using LDX composite workload feature and run the composite workload against the array




| Host IOs/sec | % Writes | %Reads | Avg I/O Size | Capacity (GB) |
|---|---|---|---|---|
| 522.1 | 0.4 | 99.6 | 19 | 256 |
| 448.5 | 0.1 | 99.9 | 16 | 256 |
| 316.6 | 5.2 | 94.8 | 19 | 256 |
| 297 | 0.9 | 99.1 | 29 | 100 |
| 235.6 | 4.8 | 95.2 | 17 | 256 |
| 220.2 | 5.6 | 94.4 | 20 | 256 |
| 201.4 | 1.5 | 98.5 | 237 | 256 |
| 165.7 | 5.1 | 94.9 | 19 | 200 |
| 91.9 | 6.2 | 93.8 | 17 | 100 |
| 90.3 | 27.7 | 72.3 | 48 | 200 |
| 70.2 | 96.9 | 3.1 | 28 | 32 |
| 68.1 | 99.6 | 0.4 | 14 | 32 |
| 7.6 | 17.8 | 82.2 | 105 | 256 |
| 6.3 | 88.5 | 11.5 | 13 | 10 |
| 2.8 | 98.4 | 1.6 | 347 | 256 |
| 1.5 | 17.7 | 82.3 | 2 | 33 |
| 0.1 | 11.5 | 88.5 | 4 | 33 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 2.5 | 97.5 | 0 | 33 |

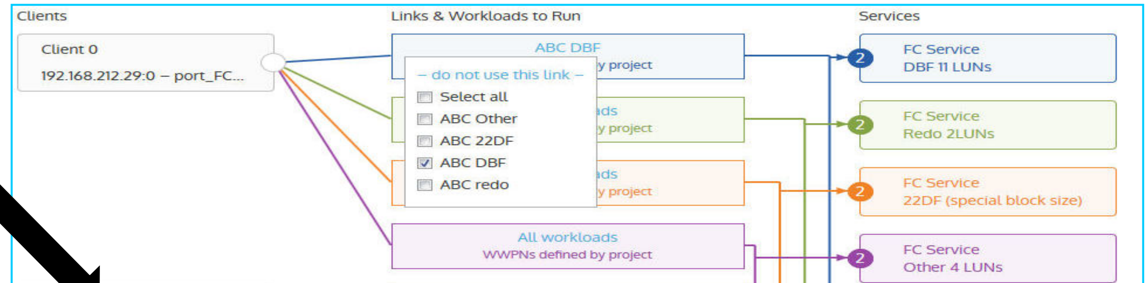


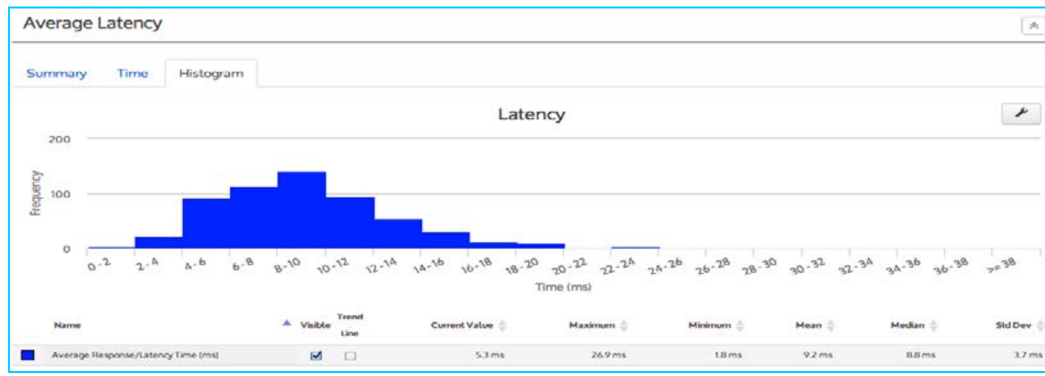

SDC 15

# Sample Use Case - Analysis

3. View distribution of latency (and other figures) as related to the mission critical database workload

| Host IOs/sec | % Writes | %Reads | Avg I/O Size | Capacity (GB) |
|---|---|---|---|---|
| 522.1 | 0.4 | 99.6 | 19 | 256 |
| 448.5 | 0.1 | 99.9 | 16 | 256 |
| 316.6 | 5.2 | 94.8 | 19 | 256 |
| 297 | 0.9 | 99.1 | 29 | 100 |
| 235.6 | 4.8 | 95.2 | 17 | 256 |
| 220.2 | 5.6 | 94.4 | 20 | 256 |
| 201.4 | 1.5 | 98.5 | 237 | 256 |
| 165.7 | 5.1 | 94.9 | 19 | 200 |
| 91.9 | 6.2 | 93.8 | 17 | 100 |
| 90.3 | 27.7 | 72.3 | 48 | 200 |
| 70.2 | 96.9 | 3.1 | 28 | 32 |
| 68.1 | 99.6 | 0.4 | 14 | 32 |
| 7.6 | 17.8 | 82.2 | 105 | 256 |
| 6.3 | 88.5 | 11.5 | 13 | 10 |
| 2.8 | 98.4 | 1.6 | 347 | 256 |
| 1.5 | 17.7 | 82.3 | 2 | 33 |
| 0.1 | 11.5 | 88.5 | 4 | 33 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 0 | 100 | 0 | 0.5 |
| 0 | 2.5 | 97.5 | 0 | 33 |




SDC 15

# 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 is building upon process methodology developed by the SNIA SSS TWG and Green TWG

# Summary

# Summary

- Flash Arrays are unlike disk-based arrays
- Data reduction dramatically changes performance characteristics
- Testing an AFA requires:
  - Real-world access patterns
  - Rich data content
  - Testing that accounts for enterprise features

# Load DynamiX Benefits

- **Optimize Storage Investment**
  - Eliminate over/under-provisioning, or stove-piping, by aligning your workload requirements to deployment decisions
- **Mitigate Risk**
  - Identify issues before deployment by testing at extreme scale and worst-case conditions
- **Innovate with Confidence**
  - Adopt the latest storage technologies without the fear of impacting application performance

**"If you can't validate technology before it's deployed into production, then you're flying blind."**

**Julia Palmer**
Performance Engineering Manager