SDC

STORAGE DEVELOPER CONFERENCE

SNIA SANTA CLARA, 2017

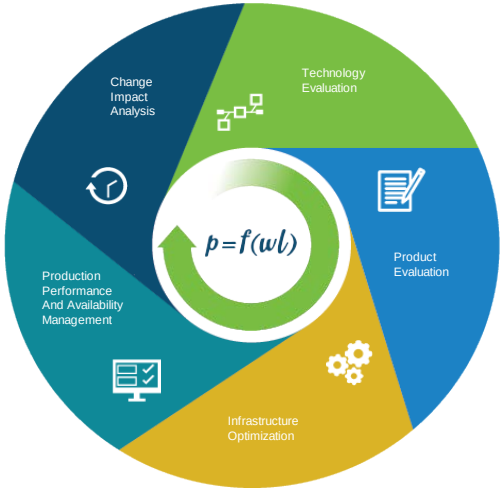
Getting it Right: Testing Storage Arrays The Way They'll be Used: A Methodology

Peter Murray

Virtual Instruments

Agenda

- Introduction
- Application Emulation
  - Locality
    - Hot spots/bands
  - Realistic access patterns
  - Bursts
  - Data Content
- Combining Applications
- Summary



Introduction

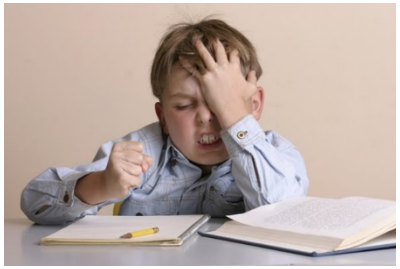
- Solid state storage arrays are mainstream
- Effectively testing solid state arrays is unlike testing disk arrays
  - New features add capability and need to be tested
  - Emulating applications shows how a storage array performs in production
  - Corners testing and benchmarks are not enough
- Emulating applications is best for performance testing

Application Emulation

- Application streams have common characteristics:
  - Locality
  - Access Patterns
  - Block Sizes
  - Data content patterns
  - Bursts
- These items critical to understanding performance in production

Writing to Solid State Arrays

- Solid state memory has a limited number of write cycles
- Therefore, modern solid state storage arrays avoid writing
- Write access is very different than read access
- Flash write access time is implementation dependent
  - Sequential writing may be impacted
  - Random writing can impact garbage collection
- Data reduction processing may require post-processing
  - But typically does not affect write speed



Writing is Hard

Realistic Access

- Testing should emulate real applications
  - Should avoid uniform random write distribution
  - Should use multiple block sizes
    - Both result in unrealistic access patterns that skew towards systems that maintain large amounts of reserve flash memory
- Methodology should include testing in the presence of:
  - Backups
  - Snapshots
  - Replication
  - Periodic access

Realistic Access Patterns

- Access pattern factors:
  - Write/read ratios
  - Random/sequential access ratios
  - Access pattern drift
  - Block sizes
  - Alternate paths

Locality

- Locality is a realistic application component
  - Even with blended applications
- Some storage arrays use locality to determine where and when data gets written
- Locality defines:
  - Where data is written or read – spatial locality
  - When data is written or read – temporal locality
  - Hot spots/hot bands represent locality
- Testing without locality does not stress an array as it will be in production

Locality Demonstration

- Shows write locality for a LUN with real-life Oracle ASM customer data
- The horizontal axis is the LUN broken into 1000 address segments.
- E.g.:
  - A bar on the left shows write hits at the beginning LUN addresses
  - A a bar on the right shows write hits at the ending LUN addresses
  - Each refresh shows the hits for 1 minute
  - The height of the bars is normalized for each refresh. This visualization is not meant to show relative intensity, but rather to show that hot spots move around and are all over.
  - Thanks to Lou Lydiksen, Pure Storage, for the data.
- The demo is here

Access Patterns

- Application access is not uniformly random
- Hot spots are storage locations accessed more frequently than others during a defined time period
  - Index files
  - Temp files
  - Logs
  - Journals
- Testing should accurately emulate data offset movement, or “Drift”, over time

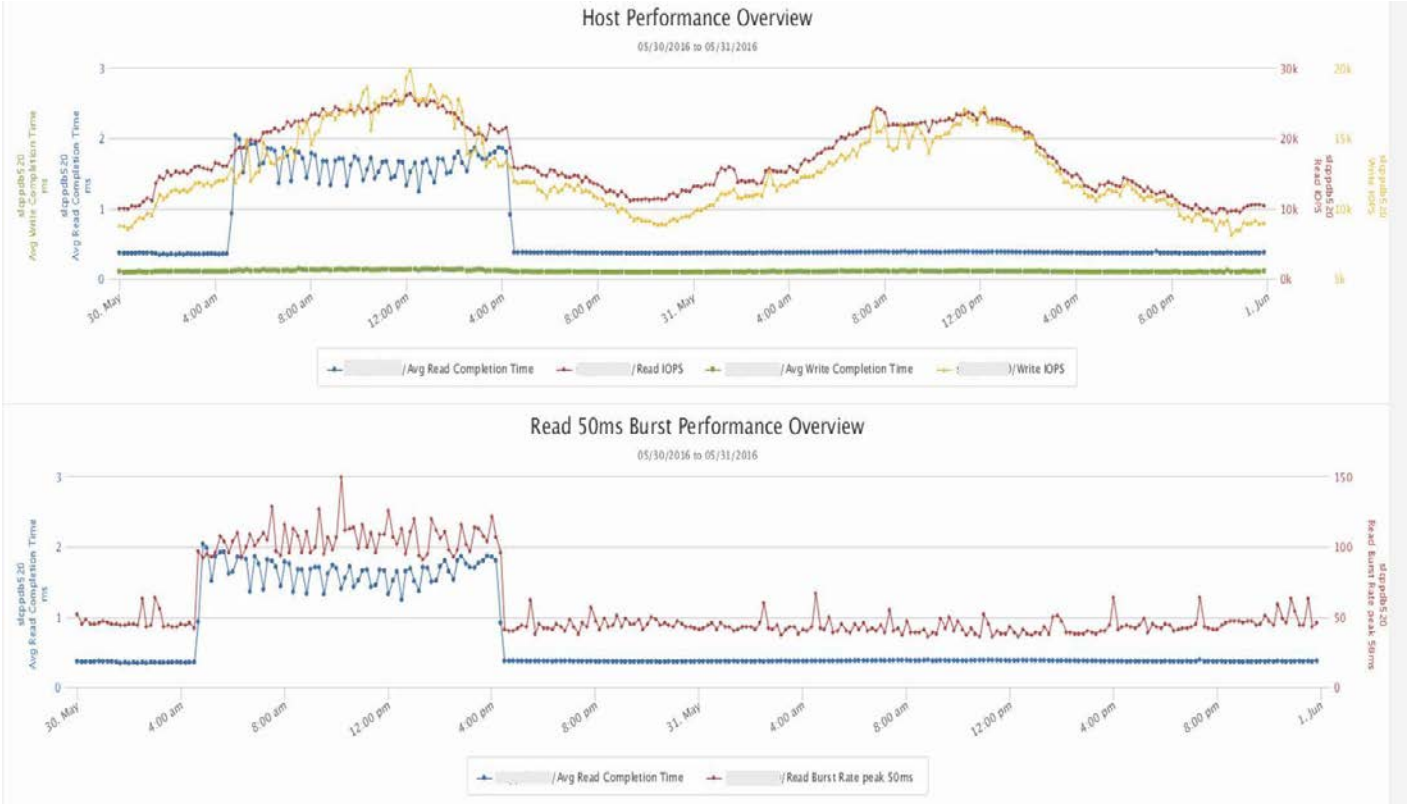
Testing Should Reflect Hot Spots and Skew

- 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
- Note: The developer of fio has written that skew is even greater than this example

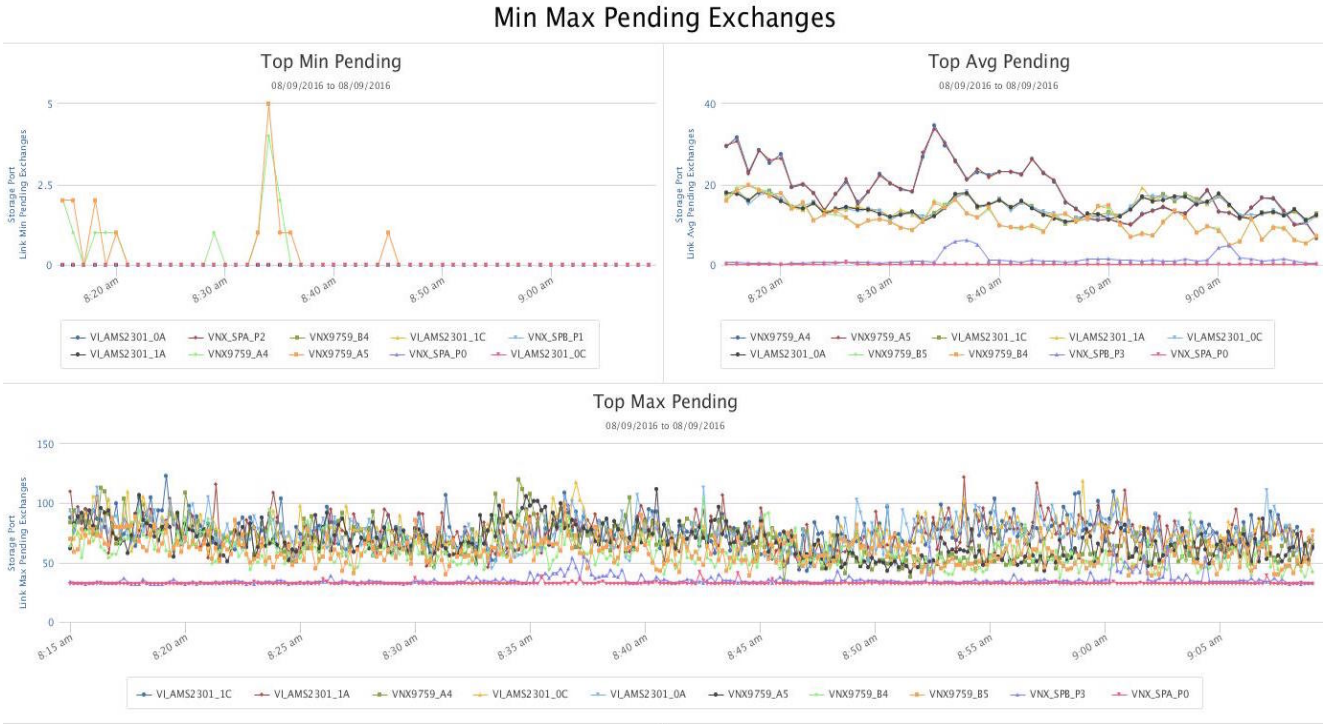
Block Sizes

- Block sizes vary by application and operation
  - 25K-35K average size reported by arrays is common
- However, applications rarely use uniform block sizes
  - Sizes vary according to operations
    - OLTP transactions typically small
    - Analytics, reporting typically larger
- Testing must include representative block sizes
  - Block sizes should be mixed to reflect applications
    - E.g. 3% 4K, 15% 8K, 20% 16K, 52% 32K, 10% 64K

Bursts: Real world issue

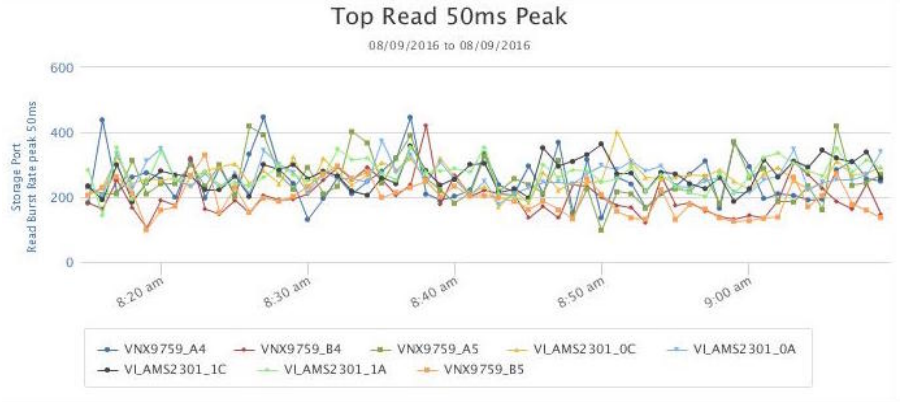


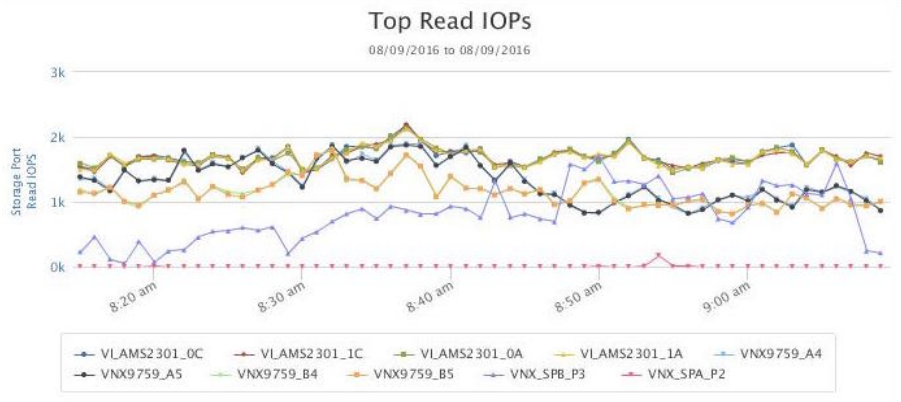
Bursts: What's Real?



For larger data centers Top Max Pending: **500-700**

Bursts: What do Real Reads Look Like?

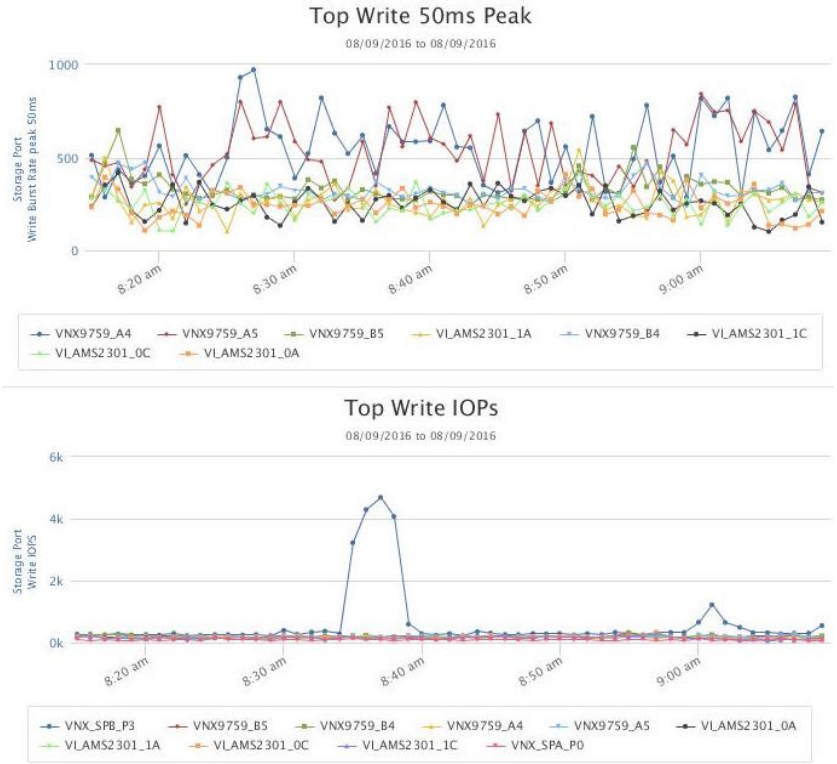




If constant would mean:
40K to 80K IOPs

The bursts are ~**40** times as high as the average

Bursts: What do Real Writes Look Like?



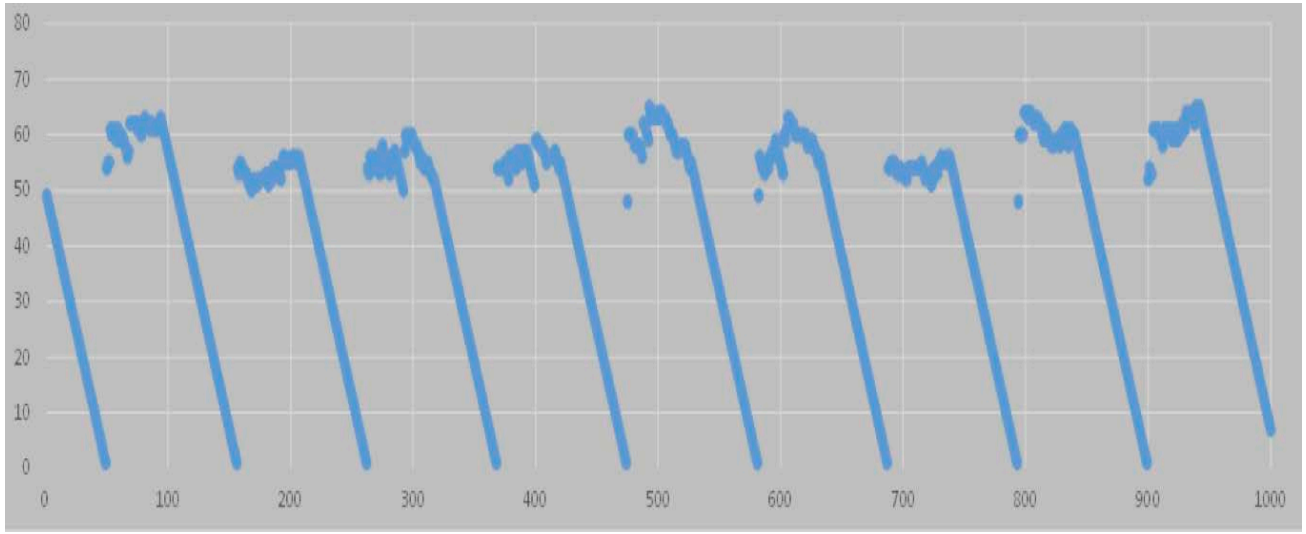
Even a bigger difference
For writes

Bursts: What's Real Writes?

Even a bigger difference
For writes

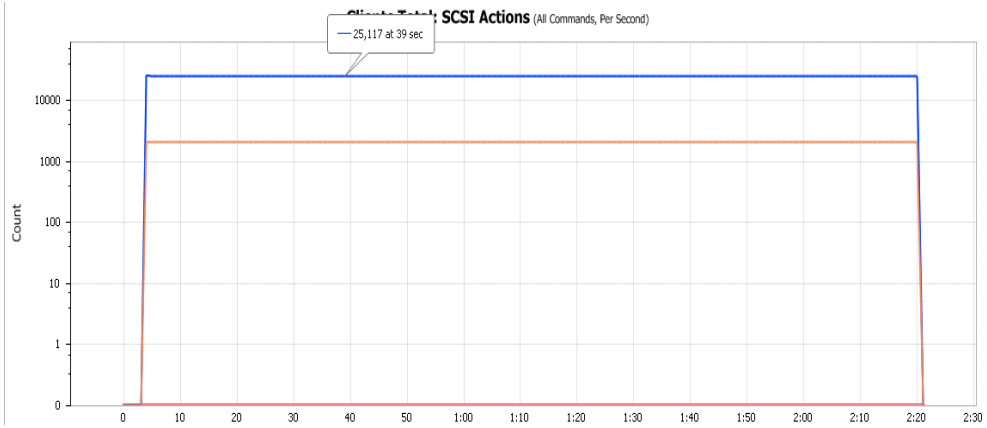
50ms bursts > 1 minute
Average IOPs

Bursts: What LDX Generates by Default

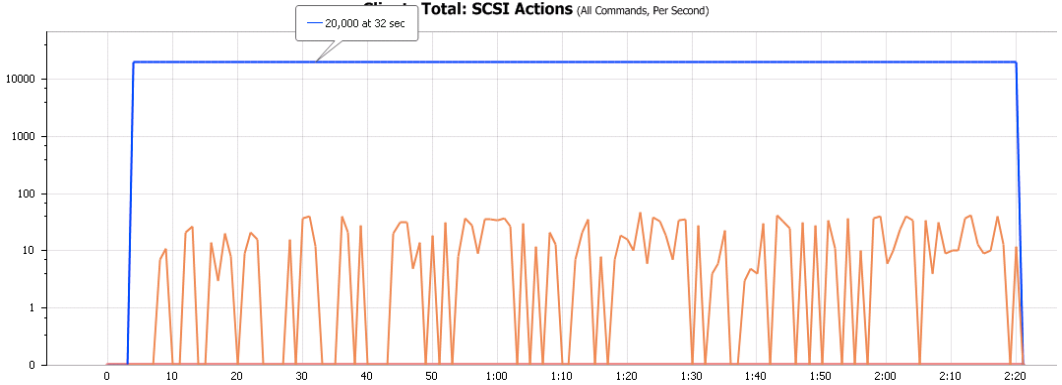


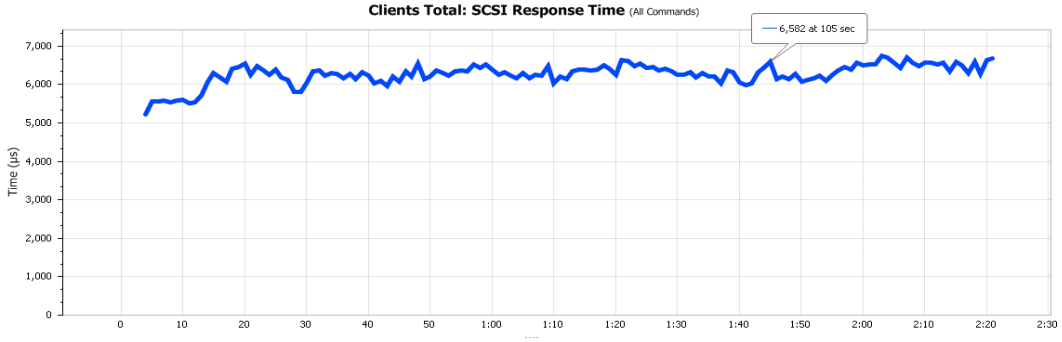
One second of data

Bursts: Which looks like

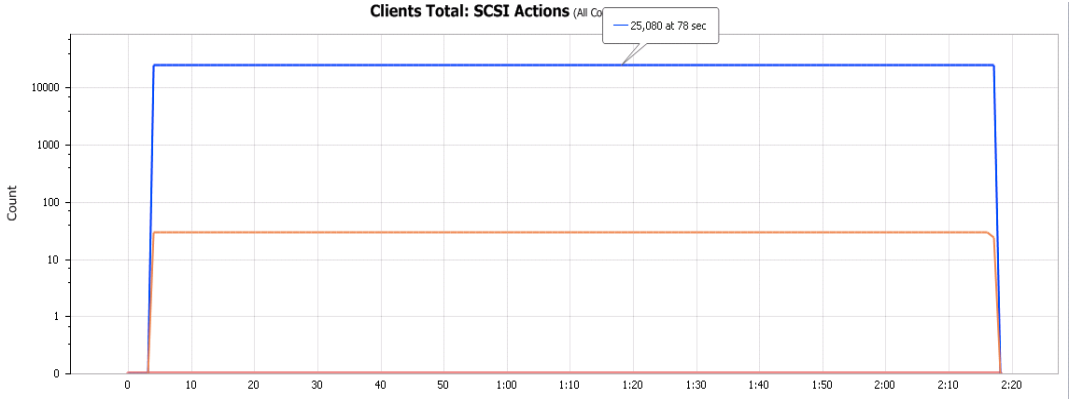


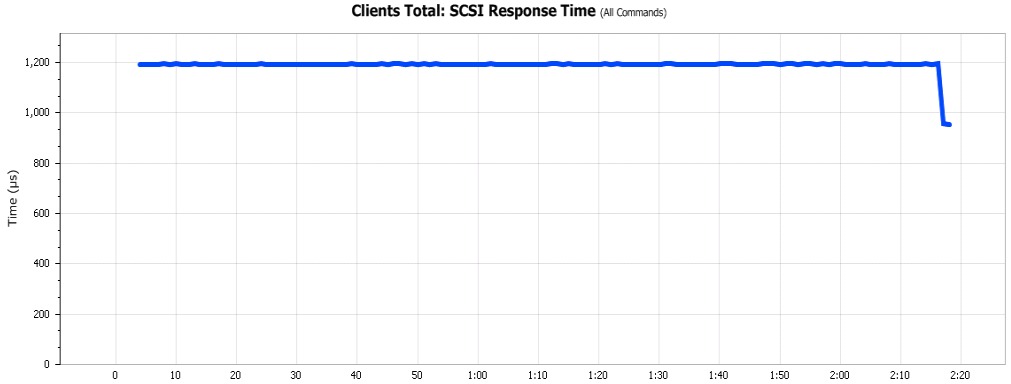
Bursts: Running at 80% of maximum





Running Without bursts





Bursts: The difference?

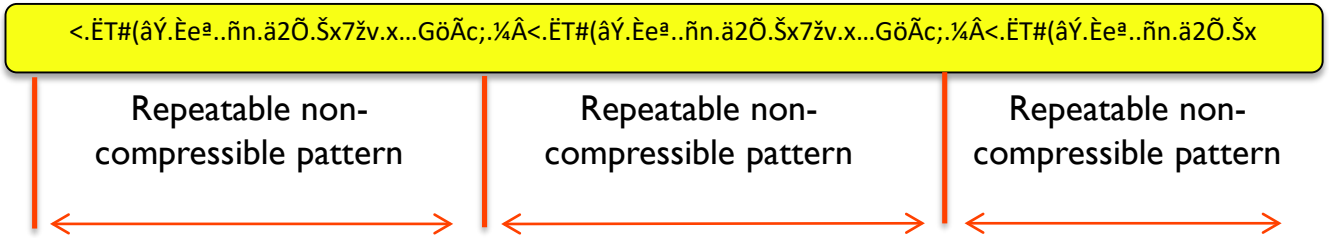
Test	IOPs	Throughput	Latency	Gap
With bursts	20K	1250 MB	6.5 ms	Real world
With bursts	25K	1569 MB	80 ms	Unacceptable latency
Without bursts	25K	1566 MB	1.2 ms	Lab-Myth

Data Content

- Modern Storage arrays use data reduction
- Data reduction saves array space
- Consists of:
  - Deduplication
  - Compression
  - Pattern reduction
- Data content patterns are a must for testing data reduction

Measuring Data Reduction

- 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

- Thread counts and queue depth
  - Tests should include increasing thread counts to find maximums for each test case
  - Should include increasing queue depth to find maximums for each test case
- Find max IOPs an array can do per thread/queue depth and total for a given number of threads and queue depth
- Increase thread count past current requirements to show how array meets future needs

Combining Applications

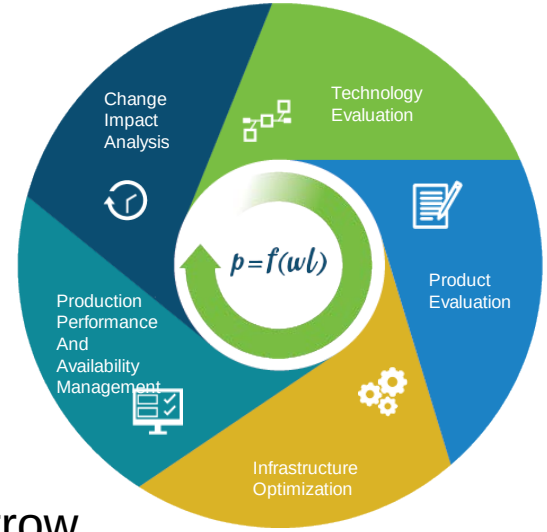
- Solid state arrays often host multiple applications
  - The use of application “stands” is decreasing
  - Should be tested together to ensure performance and reliability
- Testing should combine applications
  - Each application emulated
  - Combined applications should then be tested with snapshots / backups / replication / periodic processing
  - Care should be taken to ensure peak times are represented
- This is a work in progress
  - Methods to speed the process are needed

Testing for Expected Maximums

- Enterprise testing should ensure an array is sized properly
  - Applications usually grow over the life of an array
  - Testing should reflect peak expected use

Summary

- Application Testing is no longer a “black art”
  - No synthetic workload is perfect
  - But is the best approach available
- Customers can now see:
  - How closely a model emulates apps
  - A view of how an array operates today and tomorrow
- This new model is changing storage testing



Thank You

For more information:

peter.murray@virtualinstruments.com

www. virtualinstruments.com