Make Testing Work for You: Effective Performance Testing for Flash Storage

Training Session

Speakers / Trainers

- **Leah Schoeb**

  Manager, Solutions Reference Architecture and Performance
  Intel

- **Dennis Martin**

  Founder and President
  Demartek

- **Peter Murray**

  Principal Systems Engineer
  Load DynamiX

Agenda

1 – 1:15 pm
Introductions – All

1:15 – 2:15 pm
Flash storage testing overview – **Leah Schoeb**

2:15 – 2:30 pm
Break

2:30 – 3:30 pm
Testing Flash storage using Vdbench – **Dennis Martin**

3:30 – 4:30 pm
Testing flash storage using Load DynamiX – **Peter Murray**

4:350 – 5 pm
Q&A – All

Flash Storage Testing Overview

Leah Schoeb

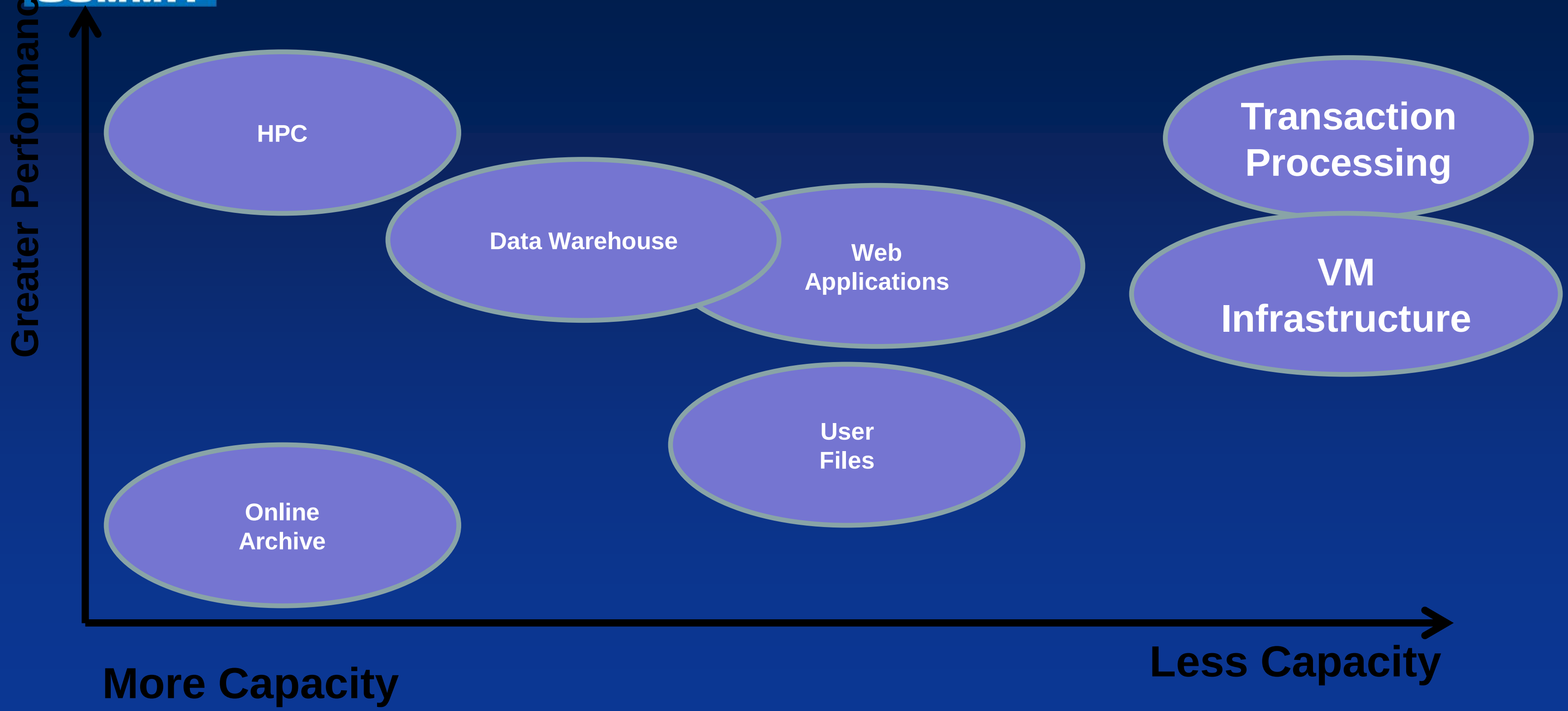
Considerations to Accurately Measure Solid State Storage Systems

Leah Schoeb

Storage Solutions and Performance Manager

Intel

Landscape for Solid State Storage



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

7

Considerations for Solid State Storage Arrays

Data Services Management	Investment Protection	Support
Data Reduction - Deduplication - Compression - Thin Provisioning	**Self-healing techniques (Reliability)**	**Hypervisor** - VMware vSphere - MS Hyper V
Replication - Local (writable) - Remote (Future)	**Hardware Redundancy (Availability)**	**Scale out and Clustering**
Management - Non-disruptive upgrades - REST APIs	**Serviceability**	**Application & OS**

Measuring Accurate Performance w/ All Flash Arrays

Problem	• Traditional IO generation tools don't work – Inadequate tool sets • Measuring new technology based on old assumptions – **Don't Do It!** • Result – **Inflated performance results**, inaccurate measurements
Flash as a unique behavior	• Not a hard disk drive
Built-in data services	• Inline data reduction technologies
Different Performance curve	• Flash arrays measure differently than traditional systems

Modern All Flash Arrays vs. HDD Arrays

Modern Flash Arrays	Traditional HDD Arrays
• Wear Leveling • Garbage collection • Metadata Management • Self-healing techniques • Inline Data deduplication • Inline Compression	• Rotational Latency • Seek Times • Mechanical parts • Controllers designed to handle HDD

Measuring w/ inline Data Reduction

Data content patterns and data streams

- Patterns written to disk as part of pre-conditioning
- Patterns presented to an array during steady state

Repeating and Non-Repeating

- random patterns
- Compressible patterns

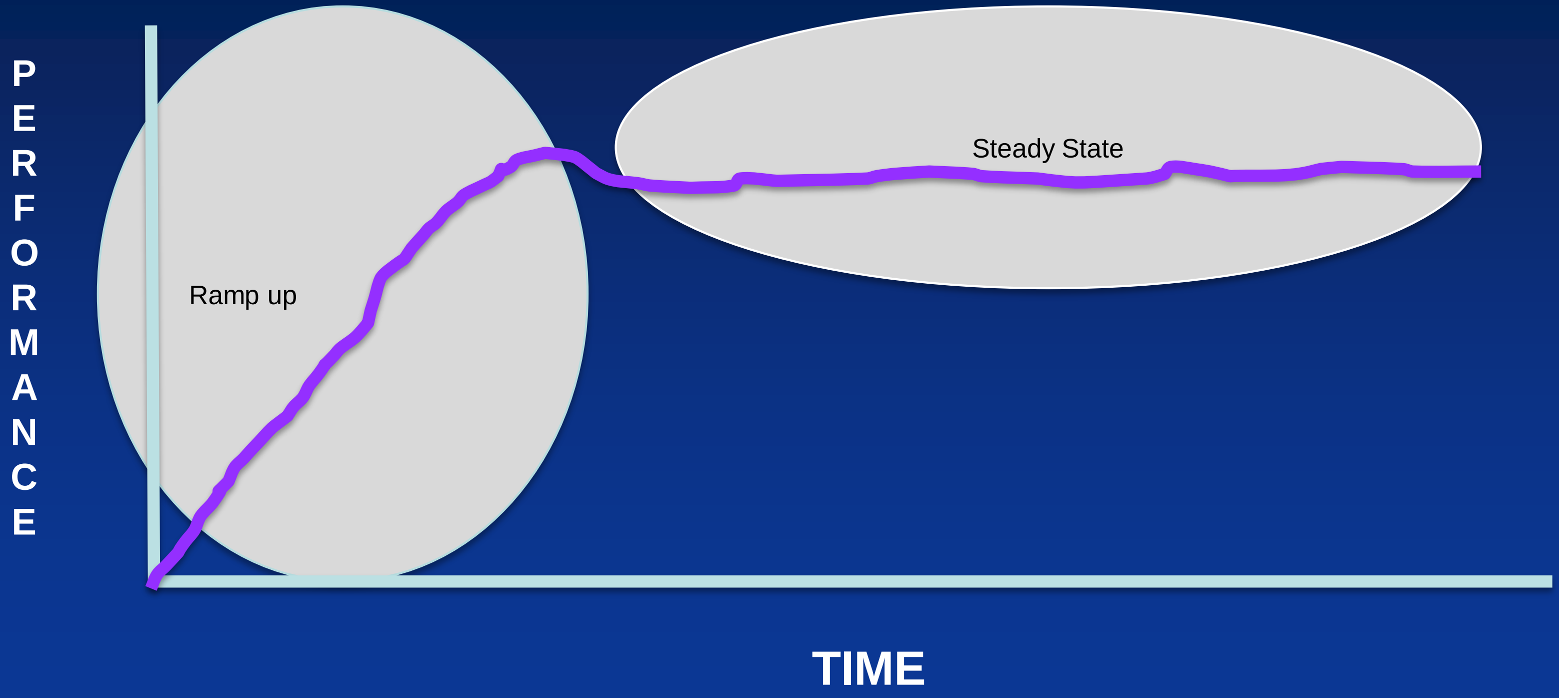
Varying pattern lengths

Most IO generators are inadequate

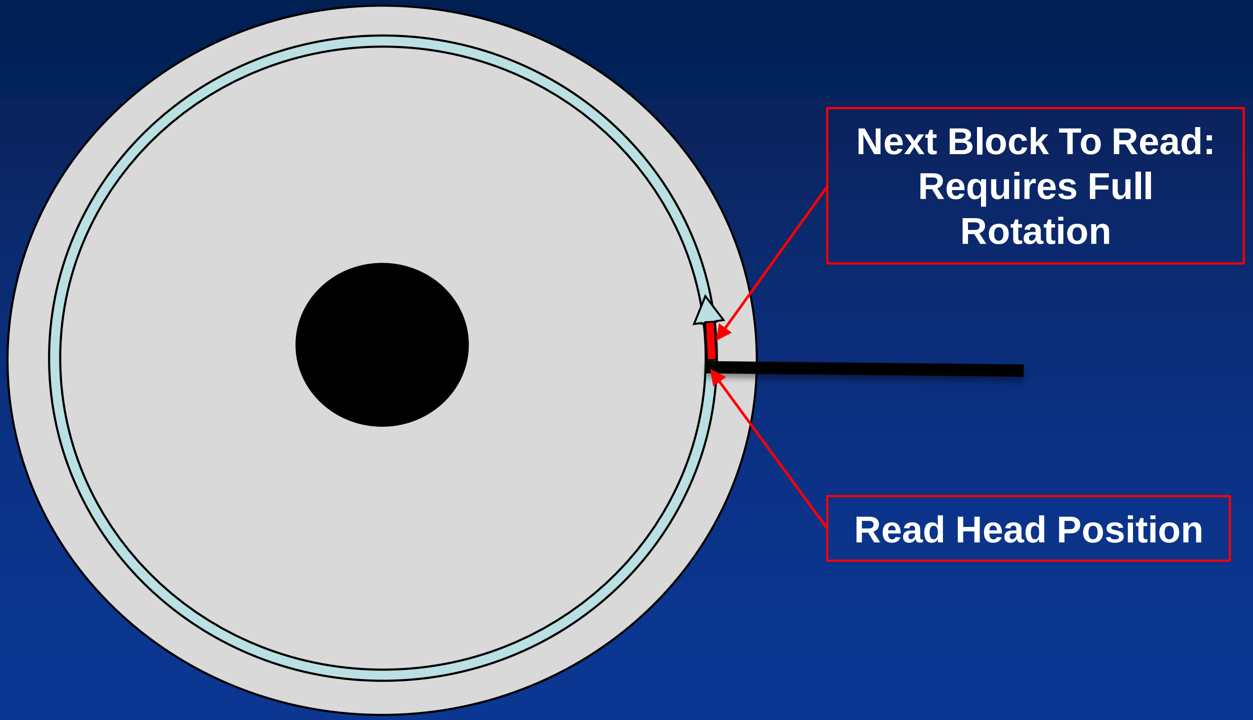
<.ËT#(âÝ.Èeª..ñn.ä2Õ.Šx7žv.x...GöÃc;.¼Â<.ËT#(âÝ.Èeª..ñn.ä2Õ.Šx7žv.x...GöÃc;.¼Â<.ËT#(âÝ.Èeª..ñn.ä2Õ.Šx

Repeatable non-compressible pattern	Repeatable non-compressible pattern	Repeatable non-compressible pattern

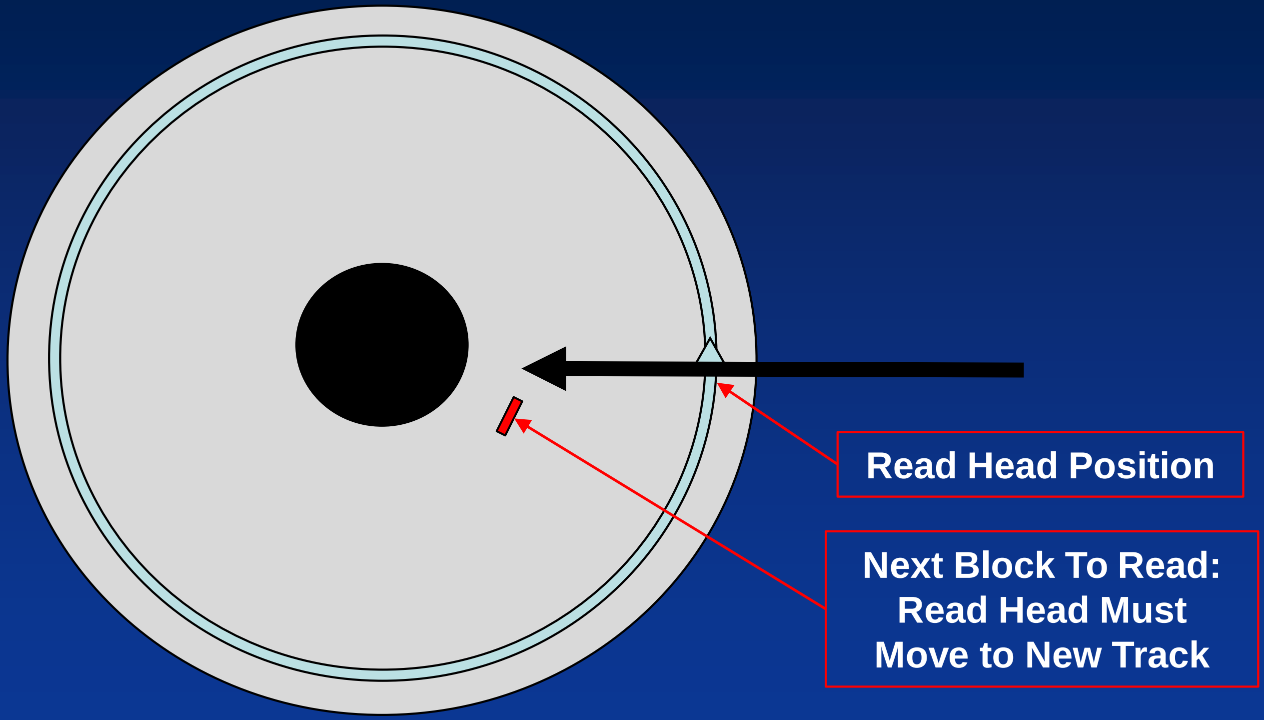
Traditional Disk Performance Curve



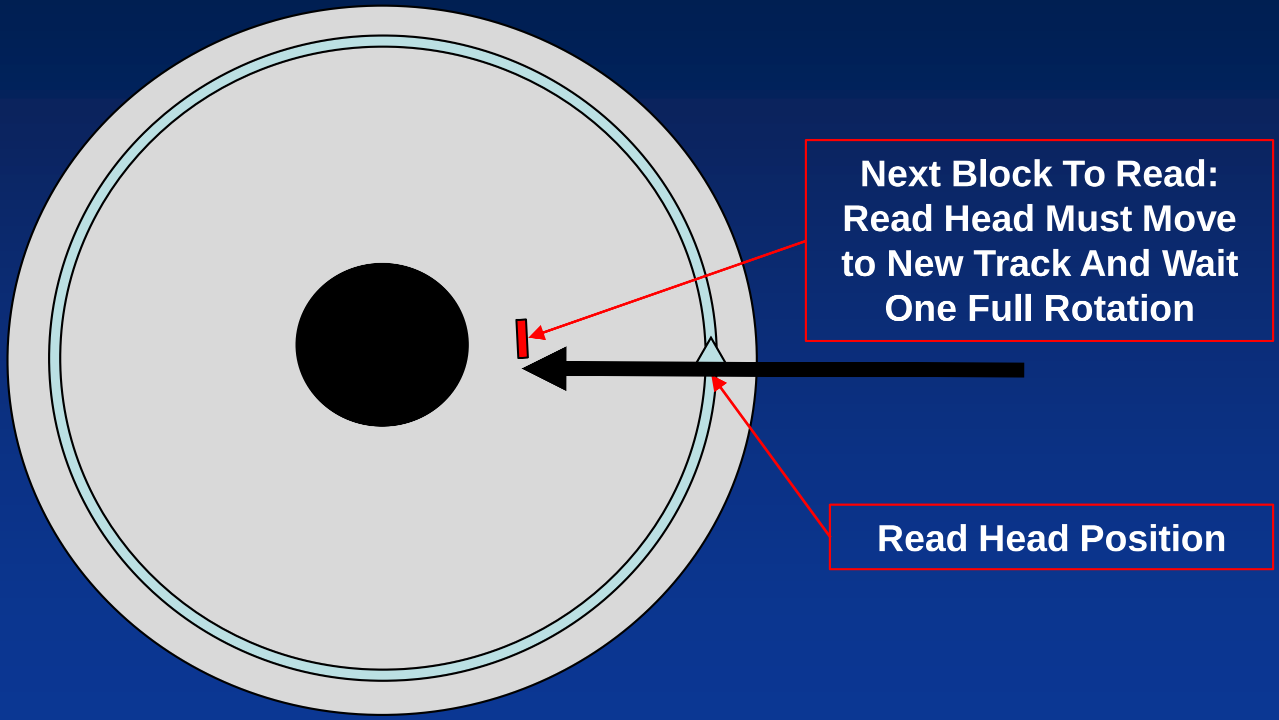
Rotational Latency

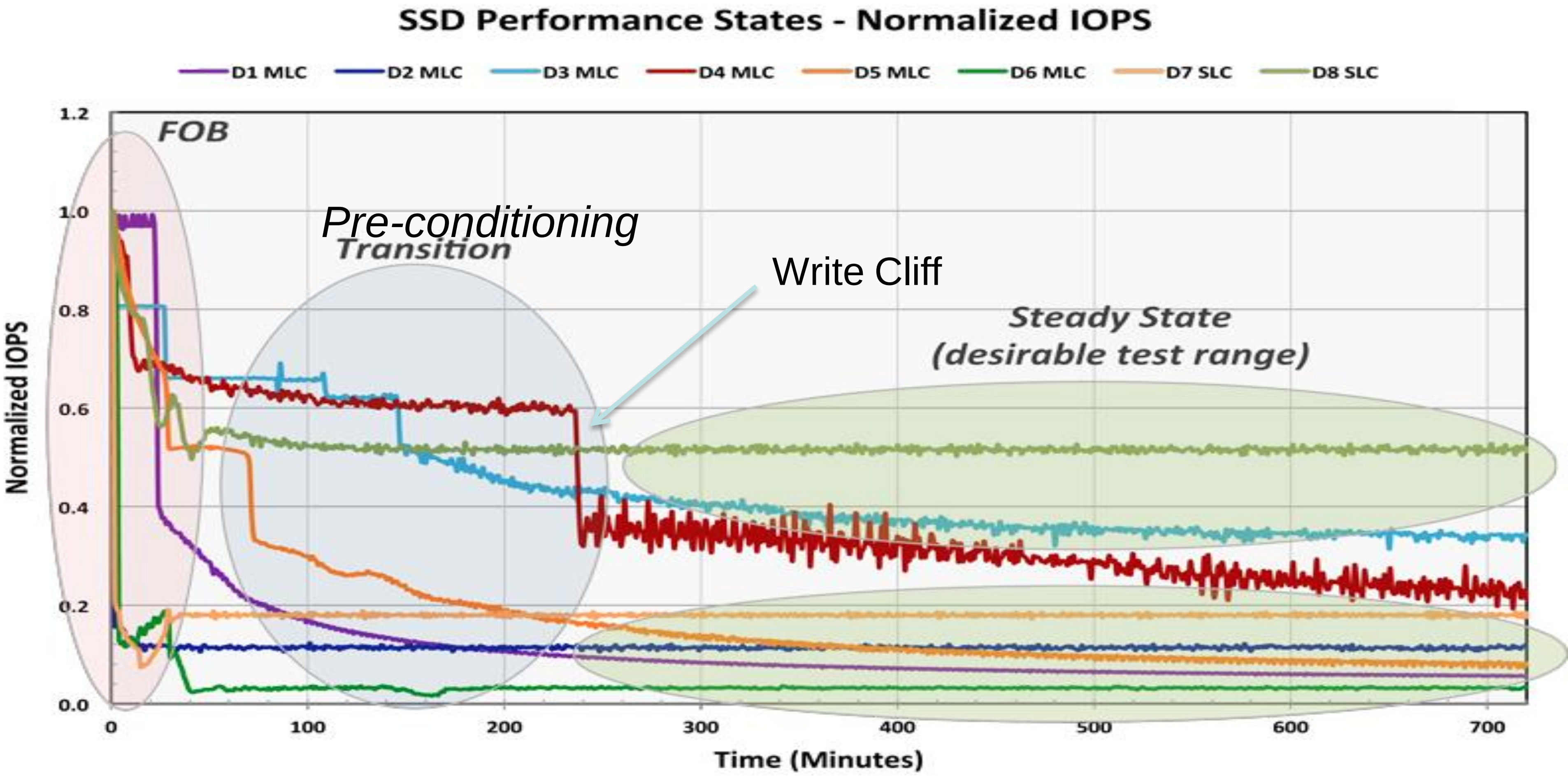


Stroke Latency



Worst-Case Latency





(SNIA SSSI Specification)

Methodology Overview

Methodology Elements

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

18

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

19

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

20

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 capacity

- Magnifying the load to test future expected loads

21

Methodology Components

Block Size

Block sizes vary by application and operation

- 25K-35K average size is common
- However, no application uses uniform block sizes
- Sizes vary according to operations

OLTP transactions typically small

Analytics, reporting typically larger

AFA methodology should reflect real access

- Single application
- IO Blender (multiple applications)
- Either model requires multiple block sizes

Should reflect application/blender access

- 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 change over time
- Called drift
  - E.g. Index file growth as transactions are processed

Hot Spot examples:

- Index Files
- Temp Files
- Logs
- Journals

Hot Spots/Bands and Drift (continued)

- 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
  - Both result in unrealistic access patterns that skew towards systems that maintain larger amounts of reserve flash memory

Methodology should include testing in the presence of:

- Backups
- Snapshots
- Replication

Complex Data Patterns

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

<.ËT#(âÝ.Èeª..ñn.ä2Õ.Šx7žv.x…GöÃc;.¼Â<.ËT#(âÝ.Èeª..ñn.ä2Õ.Šx7žv.x…GöÃc;.¼Â<.ËT#(âÝ.Èeª..ñn.ä2Õ.Šx

Repeatable non-compressible pattern | Repeatable non-compressible pattern | Repeatable non-compressible pattern

8/10/2015

Thread Count and Queue Depth

Helps find max performance for each:	• Thread count • Queue depth
Tests must find max IOPs an array can do per:	• Thread count (workers) • Queue depth (outstanding I/Os) • Combination of threads and queue depth
Increasing thread count	• past current requirements shows how array meets future needs

New SNIA Technical Working Group

Solid State Storage System Technical Working Group

(s4twg.snia.com)

Why another Solid State TWG?

- Address the unique performance behavior of SSS storage systems
- Inline-advanced features
- Measuring Performance of enterprise arrays vs. devices
- System wide housekeeping vs devices level
- Caching and DRAM tiering

Charter

Identify, develop, and coordinate systems standards to **enable accurate performance** measurement for solid state storage systems

Produce a **comprehensive set of specifications**

Drives consistency of **measurement guidelines** and messages related to solid state storage systems

Documents **system-level requirements** and shares these with other performance standards organizations

Program of Work

Develop a specification	• measuring the performance of solid state systems.
Includes **support for inline advanced storage features**	• directly impact performance and the long term behavior of the array.

Note: This will build upon process methodology developed by the SSS TWG

Current focus areas

Targeted Workload Modeling

- Database
- Server virtualization and VDI

Characteristics

- Access Patterns
- File access (structure, location, metadata, etc.)
- Data Reduction technologies (inline & post)
- Caching affects and SRAM tiering
- System wide vs device level housekeeping

Additional Areas of focus

- Defining a SSS System / SSS Taxonomy
- **Use cases**
- Include Server Based Storage
- Power (power budgets, Green TWG, SSS)
- SMI-S integration points

Workload Trace Development

Trace and Replay tools

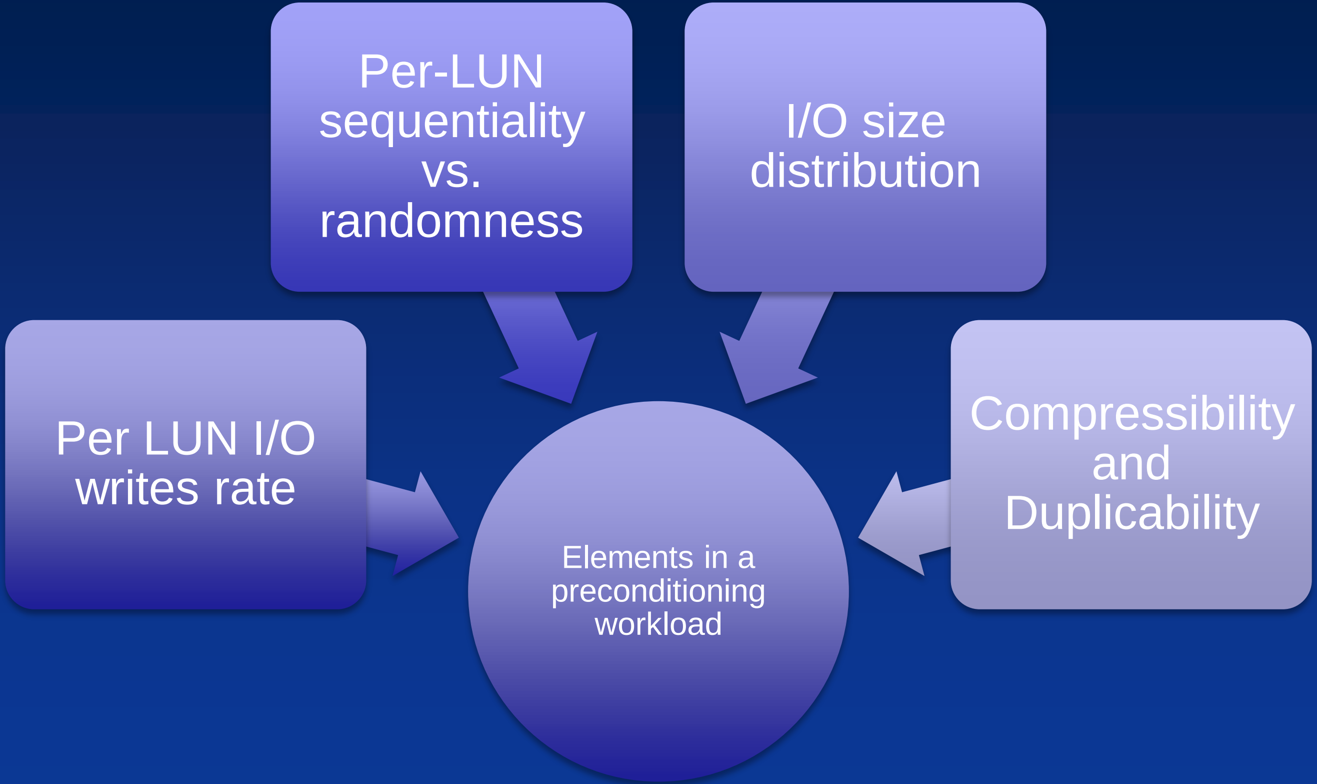
Trace Length

Creating a Model

Metrics

Workload

Workload Preconditioning



Data Preparation – First Method

Write-only Fill to 90% of available space twice	• Ensure that all physical media has been written to once
Run one test using "fill LUNs", sequential access written with non-reducible data	• Use large block sequential access pattern to fill all available physical cells with non-reducible (non-repeating random) data

Write-only using random access with non reducible data

Test minimum percentage writes for aging to be effective

Data Preparation – Second Method

Run workload dependent pre-conditioning against full array

Write to “Fill” LUNs, then “Test” LUNs

Use test LUNs	• LUNs to be part of data set • May need to trim/erase portion of non-reducible data set
Write-only to Fill LUNs twice (unused portion of available space)	• Ensure that all physical media has been written to once
First, run one test using Fill LUNs, sequential access written with non-reducible data	• Use large block sequential access pattern to fill all available physical cells with non-reducible (non-repeating random) data
Run second test on Fill LUNs	• Write-only using random access with non reducible data • Test minimum percentage writes for aging to be effective

Run workload-dependent conditioning simultaneously with workload-independent conditioning	• Run using write-only workload that uses workload dependent data and writes to the Test LUNs to fill remaining space
	  • Use reducible data set emulating an application
	  • Run 2x of available Test LUN space

Data Preparation – Third Method

Run workload-dependent data stream to prepare data set	• Run test for 10 hours • Measure whether steady state is achieved • Use steady state as defined in PTS spec
Measure how an array performs as the percentage of the array is filled?	• Ensure array is ready to capture defensible metrics • Write to both logical and physical space • Ensure every cell on an array has been written to once • Use only write component of the proposed workload
Ensure enterprise features are turned on	• Thin provisioning is turned on if available • Deduplication/Compression • Capture at various IO rates and measure variation

Workload Sources

Industry Standard workloads

- SNIA Emerald (source: Green TWG)
- SPEC SFS
- SPC

Workload Generators

- Tracing customer applications

Tiered Arrays unlike AFAs

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

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
- Testing must be fair, unbiased and repeatable

References

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

Flash Storage Testing Using Vdbench

Dennis Martin

Demartek®

Flash Storage Testing Using Vdbench

Pre-conference session

Dennis Martin, President

Demartek®

Agenda

- About Demartek
- Storage Performance Metrics
- Synthetic vs. Real-world workloads
- Vdbench

Many of the images in this presentation are clickable links to web pages or videos → 💻

About Demartek

- Industry Analysis and ISO 17025 accredited test lab
- Lab includes enterprise servers, networking & storage (DAS, NAS, SAN, 10GbE, 40GbE, 16GFC)
- We prefer to run real-world applications to test servers and storage solutions (databases, Hadoop, etc.)
- Demartek is an EPA-recognized test lab for *ENERGY STAR Data Center Storage* testin

- Website: www.demartek.com/TestLab

SNIA Emerald™ RECOGNIZED TESTER

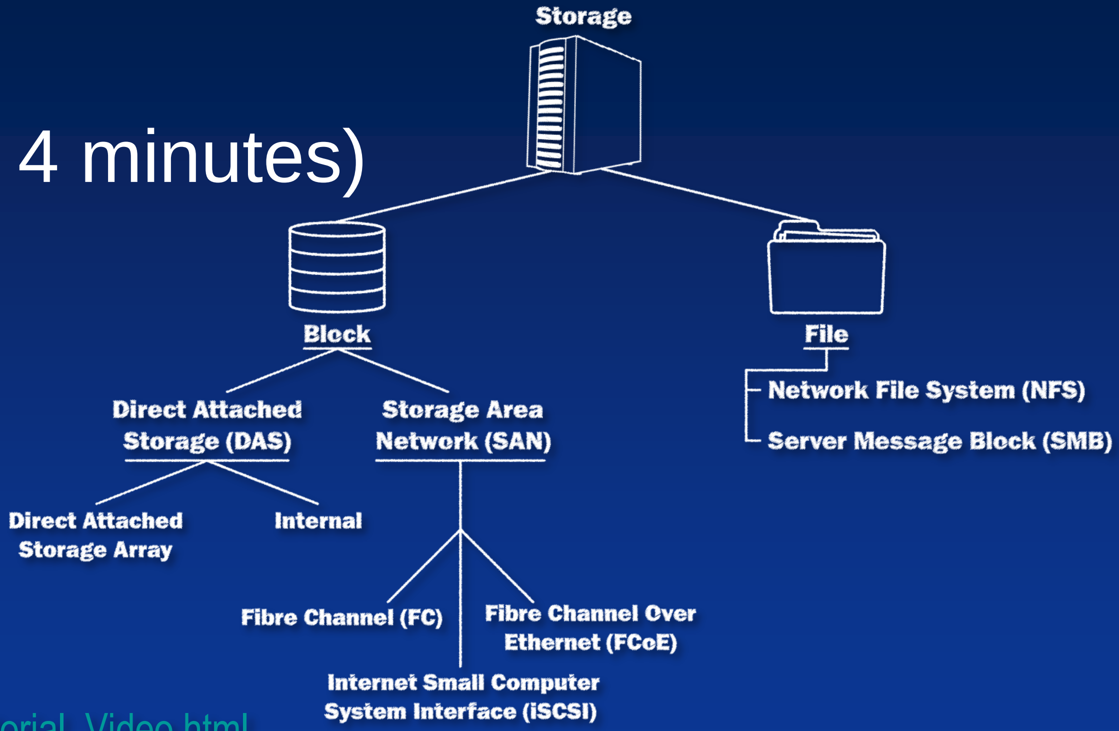
Enterprise Storage Architectures

▸ Flash Can Be Deployed In Any of These

- Direct Attach Storage (DAS)
  - Storage controlled by a single server: inside the server or directly connected to the server (“server-side”)
  - ***Block*** storage devices
- Network Attached Storage (NAS)
  - File server that sends/receives ***files*** from network clients
- Storage Area Network (SAN)
  - Delivers shared ***block*** storage over a storage network

Demartek Tutorial Videos

- Short videos (3 – 4 minutes)
- Storage Basics



http://www.demartek.com/Demartek_Tutorial_Video.html

Interface vs. Storage Device Speeds

- *Interface* speeds are generally measured in bits per second, such as megabits per second (Mbps) or gigabits per second (Gbps).
  - Lowercase “b”
  - Applies to Ethernet, Fibre Channel, SAS, SATA, etc.
- *Storage device* and system speeds are generally measured in bytes per second, such as megabytes per second (MBps) or gigabytes per second (GBps).
  - Uppercase “B”
  - Applies to devices (SSDs, HDDs) and PCIe, NVMe

Storage Interface Comparison

- Demartek Storage Interface Comparison reference page
  - Search engine: *Storage Interface Comparison*
  - Includes new interfaces such as 25GbE, 32GFC, Thunderbolt 3

http://www.demartek.com/Demartek_Interface_Comparison.html

Storage Performance Metrics

Storage Performance Metrics

▸ IOPS & Throughput

- IOPS
  - Number of Input/Output (I/O) requests per second
- Throughput
  - Measure of bytes transferred per second (MBps or GBps)
  - Sometimes also referred to as “Bandwidth”
- Read and Write metrics are often reported separately

Storage Performance Metrics

▸ Latency

- Latency
  - Response time or round-trip time, generally measured in milliseconds (ms) or microseconds (µs)
  - Sometimes measured as seconds per transfer
  - Time is the numerator, therefore lower latency is faster
- Latency is becoming an increasingly important metric for many real-world applications
- Flash storage provides much lower latency than hard disk or tape technologies, frequently < 1 ms

I/O Request Characteristics

▸ Block size

- Block size is the size of each individual I/O request
  - Minimum block size for flash devices is 4096 bytes (4KB)
  - Minimum block size for HDDs is 512 bytes
    - Newer HDDs have native 4KB sector size (“Advanced Format
  - Maximum block size can be multiple megabytes
- Block sizes are frequently powers of 2
  - Common: 512B, 1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB, 128KB, 256KB, 512KB, 1MB

I/O Request Characteristics

▸ Queue Depth

- Queue Depth is the number of outstanding I/O requests awaiting completion
  - Applications can issue multiple I/O requests at the same time to the same or different storage devices
- Queue Depths can get temporarily large if
  - The storage device is overwhelmed with requests
  - There is a bottleneck between the host CPU and the storage device

I/O Request Characteristics

▸ Access Patterns: Random vs. Sequential

- Access patterns refers to the pattern of specific locations or addresses (logical block addresses) on a storage device for which I/O requests are made
  - Random – addresses are in no apparent order (from the storage device viewpoint)
  - Sequential – addresses start at one location and access several immediately adjacent addresses in ascending order or sequence
- For HDDs, there is a significant performance difference between random and sequential I/O

I/O Request Characteristics

▸ Read/Write Mix

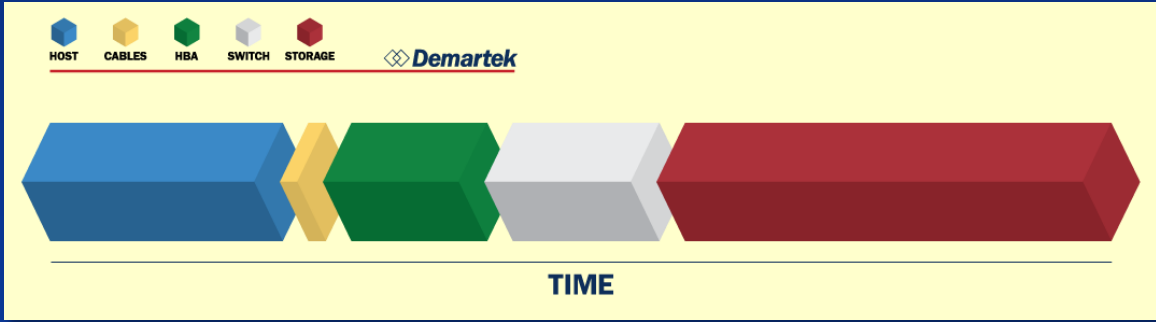
- The read/write mix refers to the percentage of I/O requests that are read vs. write
  - Flash storage devices are relatively more sensitive to the read/write mix than HDDs due to the physics of NAND flash writes
  - The read/write mix percentage varies over time and with different workloads

Storage Performance Measurement

▸ Multiple Layers

- There are many places to measure storage performance, including software layers and hardware layers
  - Multiple layers in the host server, storage device and in between

Latency example in a SAN



Synthetic vs. Real-world Workloads

Synthetic Workloads

▸ Purpose

- Synthetic workload generators allow precise control of I/O requests with respect to:
  - Read/write mix, block size, random vs. sequential & queue depth
- These tools are used to generate the *"hero numbers"*
  - 4KB 100% random read, 4KB 100% random write, etc.
  - 256KB 100% sequential read, 256KB 100% sequential write, etc.
- Manufacturers advertise the hero numbers to show the top-end performance in the corner cases
  - Demartek also sometimes runs these tests

Synthetic Workloads

▸ Examples

- Several synthetic I/O workload tools:
  - Diskspd, fio, IOmeter, IOzone, SQLIO, Vdbench, others
- Some of these tools have compression, data de-duplication and other data pattern options
- Demartek has a reference page showing the data patterns written by some of these tools
  - http://www.demartek.com/Demartek_Benchmark_Output_File_Formats.html

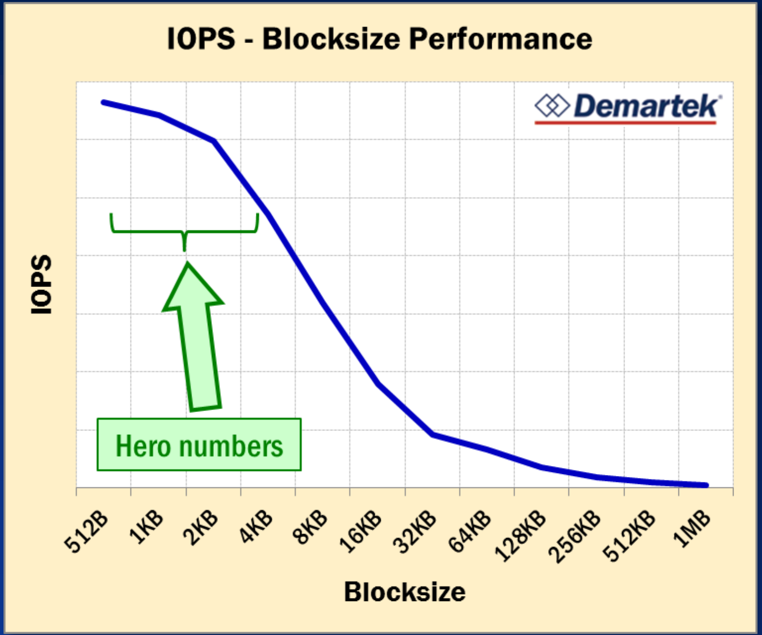
Real-world Workloads

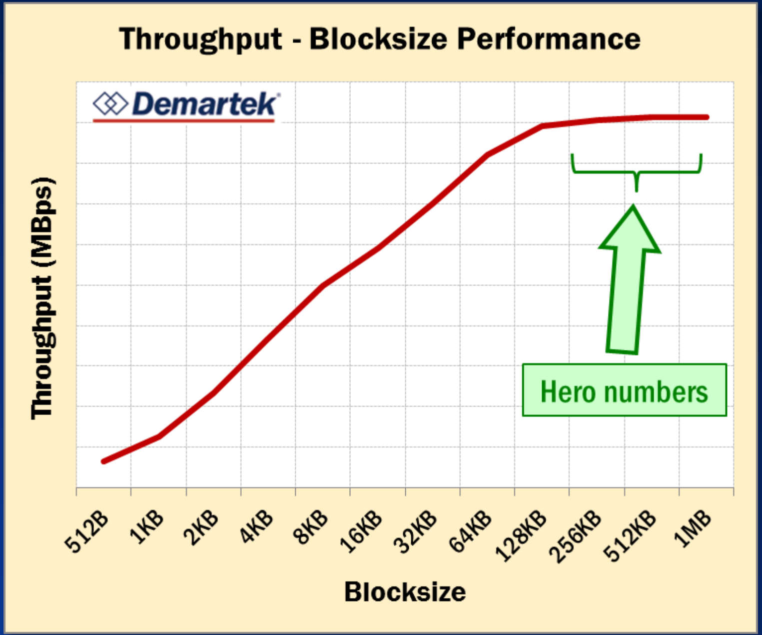
- Use variable levels of compute, memory and I/O resources as the work progresses
  - May use different and multiple I/O characteristics simultaneously for I/O requests (block sizes, queue depths, read/write mix and random/sequential mix)
- Many applications capture their own metrics such as database transactions per second, etc.
- Operating systems can track physical and logical I/O metrics
- End-user customers have these applications

Real-world Workload Types

- Transactional (mostly random)
  - Generally smaller block sizes (4KB, 8KB, 16KB, etc.)
  - Emphasis on the number of I/O’s per second (IOPS)
- Streaming (mostly sequential)
  - Generally larger block sizes (64KB, 256KB, 1MB, etc.)
  - Emphasis on throughput (bandwidth) measured in Megabytes per second (MBps)
- Latency is affected differently by different workload types

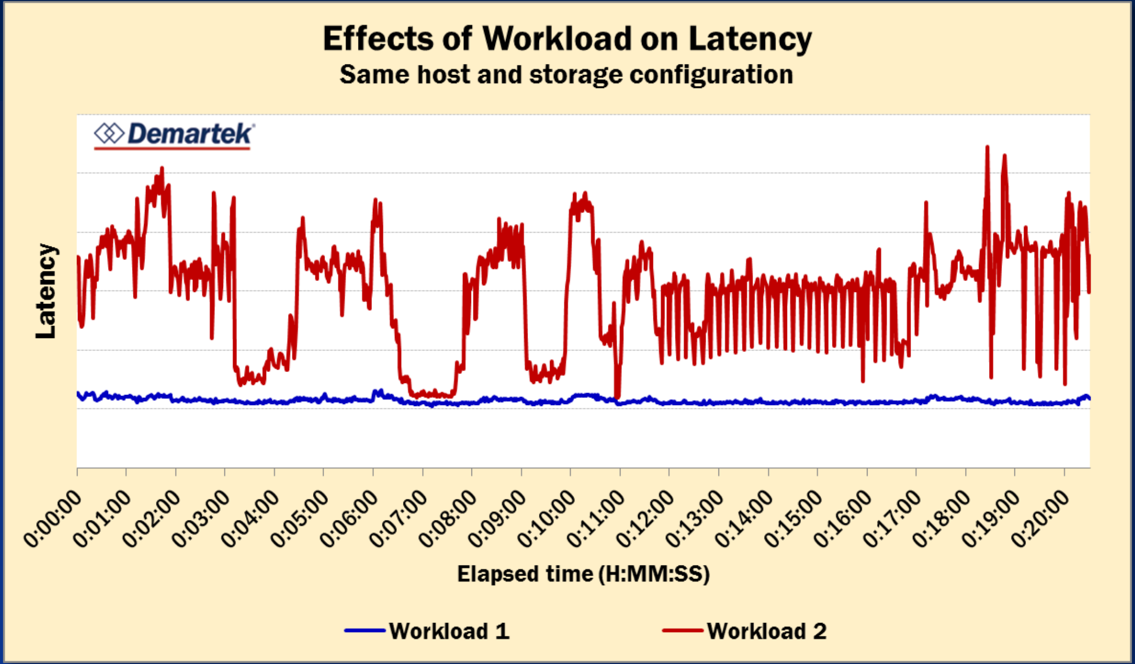
Generic IOPS and Throughput Results





These performance curves generally apply to network and storage performance

Generic Latency Results



The nature of each workload has a large impact on latency

Vdbench

Vdbench

- Vdbench is a storage I/O workload generator owned by Oracle corporation
  - http://www.oracle.com/technetwork/server-storage/vdbench-downloads-1901681.html
- Written in Java
  - Runs on a variety of Linux, UNIX and Windows systems
- Supports a variety of I/O workload parameters
  - Can be targeted at raw LUNs or volumes with file systems
- Used in SNIA Emerald and EPA ENERGY STAR Data Center Storage test specifications for block I/O testing

Vdbench Capabilities

- The *Vdbench Users Guide version 5.03* is available for download
- Vdbench has a large number of features and capabilities
- This example (and lab) will work with only a few basics

Vdbench Definitions

- Storage Definition (SD)

```
sd=sd1,lun=i:\test1.txt,size=50m
```

- Workload Definition (WD)

```
wd=wd_rnd,sd=sd1,seekpct=rand
wd=wd_seq,sd=sd1,seekpct=0,streams=2
```

Vdbench Run Definition

- Run definition

```
# Random writes test phase
rd=rd_rw_final,wd=wd_rnd,rdpct=0,xfersize=8k,th=4
# Random reads test phase
rd=rd_rr_final,wd=wd_rnd,rdpct=100,xfersize=8k,th=4
# Sequential write test phase
rd=rd_sw_final,wd=wd_seq,rdpct=0,xfersize=256K,th=4
# Sequential read test phase
rd=rd_sr_final,wd=wd_seq,rdpct=100,xfersize=256K,th=4
```

Command script

- My tests run in a command script that uses variables (script number and test run number) passed to it. The final command that is issued looks like the line below.

```
vdbench -f FMS_vdbench_script_01.txt -o FMS_output_01_Test_02
```

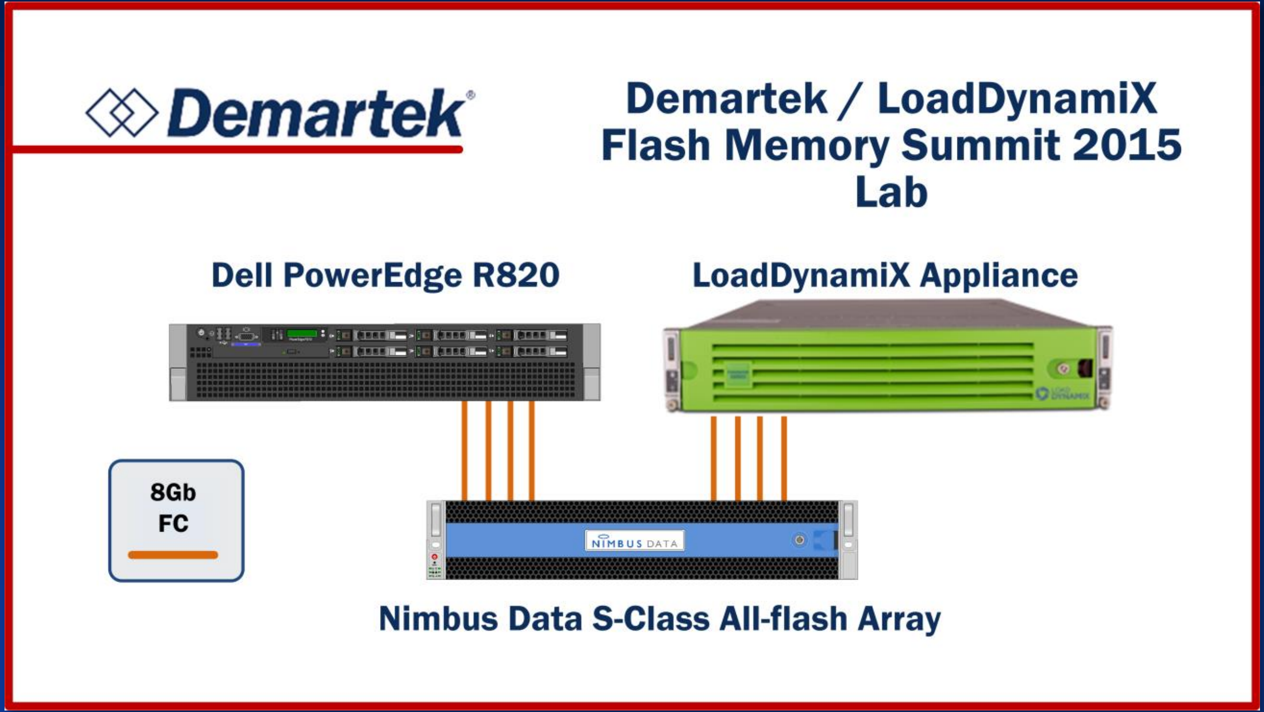
- -f option: points to the workload parameter file name
- -o option: name of output directory

Vdbench outputs

- The output directory contains several files
- Summary.html is the main output file that has links to most of the others

- DEMO of output files from previous run
- DEMO of live system

FMS 2015 Demo Lab Configuration



Thank You!

Demartek®

SUBSCRIBE TO THE DEMARTEK NEWSLETTER

Demartek public projects and materials are announced on a variety of social media outlets. Follow us on any of the above.

Sign-up for the Demartek monthly newsletter, *Demartek Lab Notes*.
www.demartek.com/newsletter

Flash Storage Testing Using Load DynamiX

Peter Murray

Agenda

1. The Challenge
2. Current Approaches
3. Model workloads & Demo
   - Use performance metrics from an existing application
   - Create a realistic workload profile
   - Simulate the production application using the Composite Workload feature of LDX Enterprise
   - Review performance metrics; analyze results
4. Summary and Q&A

Your Oracle Application Challenges

- Constant pressure to **meet performance and availability SLAs**
- Quickly evaluate, deploy and **de-risk new application rollouts**
- **Troubleshoot storage-related problems** while “in flight”
- **Implement storage-related changes** (updates/upgrades) without compromising performance and availability

Who is Load DynamiX?

Leader in storage performance validation

Mission

Provide actionable insight into storage infrastructure behavior to assure performance & optimize cost

Product Suite

Load DynamiX Enterprise test management platform combined with load generation appliance

;;

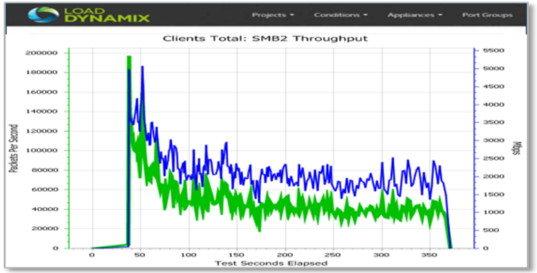
The Load DynamiX Solution

Load DynamiX Enterprise software

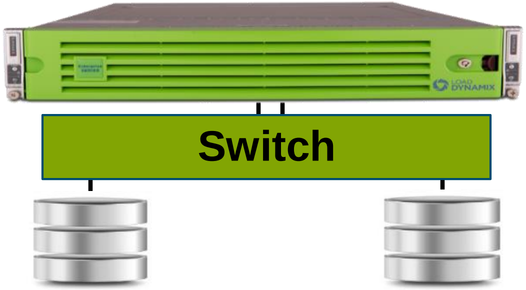
ORACLE®

Workload Modeling & Performance Profiling

Performance Analytics

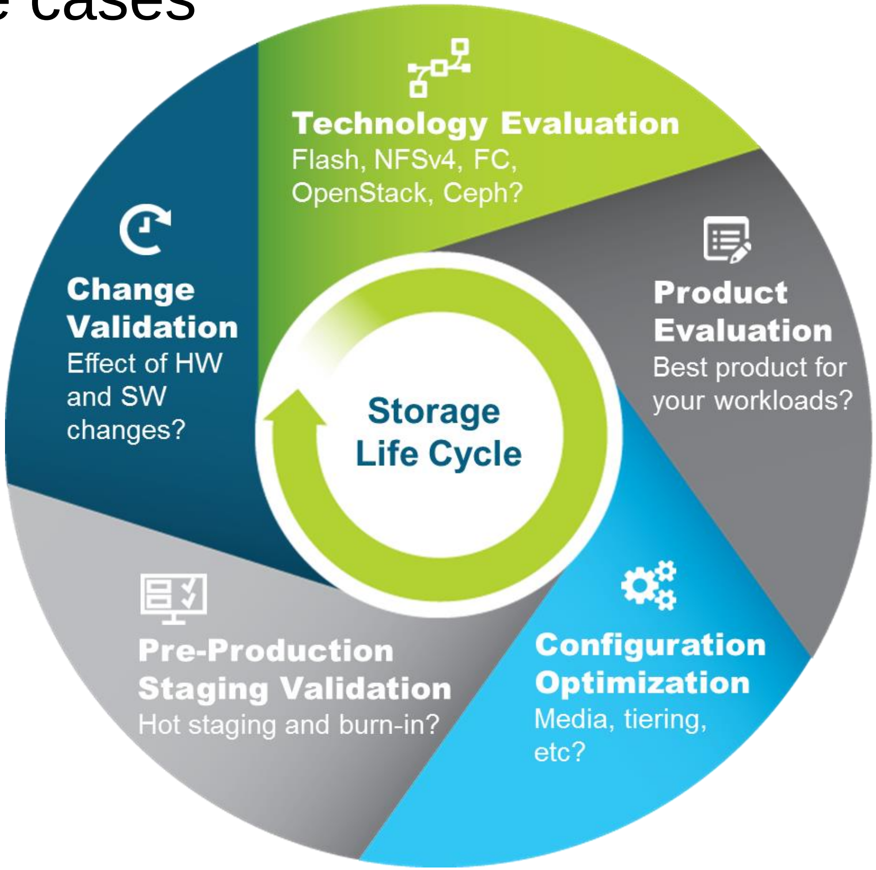


Load DynamiX Performance Validation Appliance



Why IT Organizations Use Load DynamiX

Use cases



Application Modeling

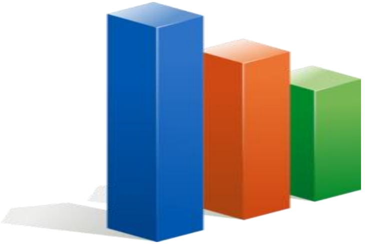
Load DynamiX Enterprise

1. Use performance metrics from an existing application
2. Create a realistic workload profile
3. Simulate the production application using the Composite Workload feature of LDX Enterprise
4. Review performance metrics; analyze results

Methodology: Workload Modeling

1 **Create a workload model***

PRODUCTION STATS
(Perfstats, .nar, .btp,
NFSstat, UniSphere, etc)



ACCURATE, REALISTIC WORKLOAD MODEL

***DB apps do not present a single I/O profile**

I/O Metrics via Storage Monitoring Tool

Via existing app on storage array

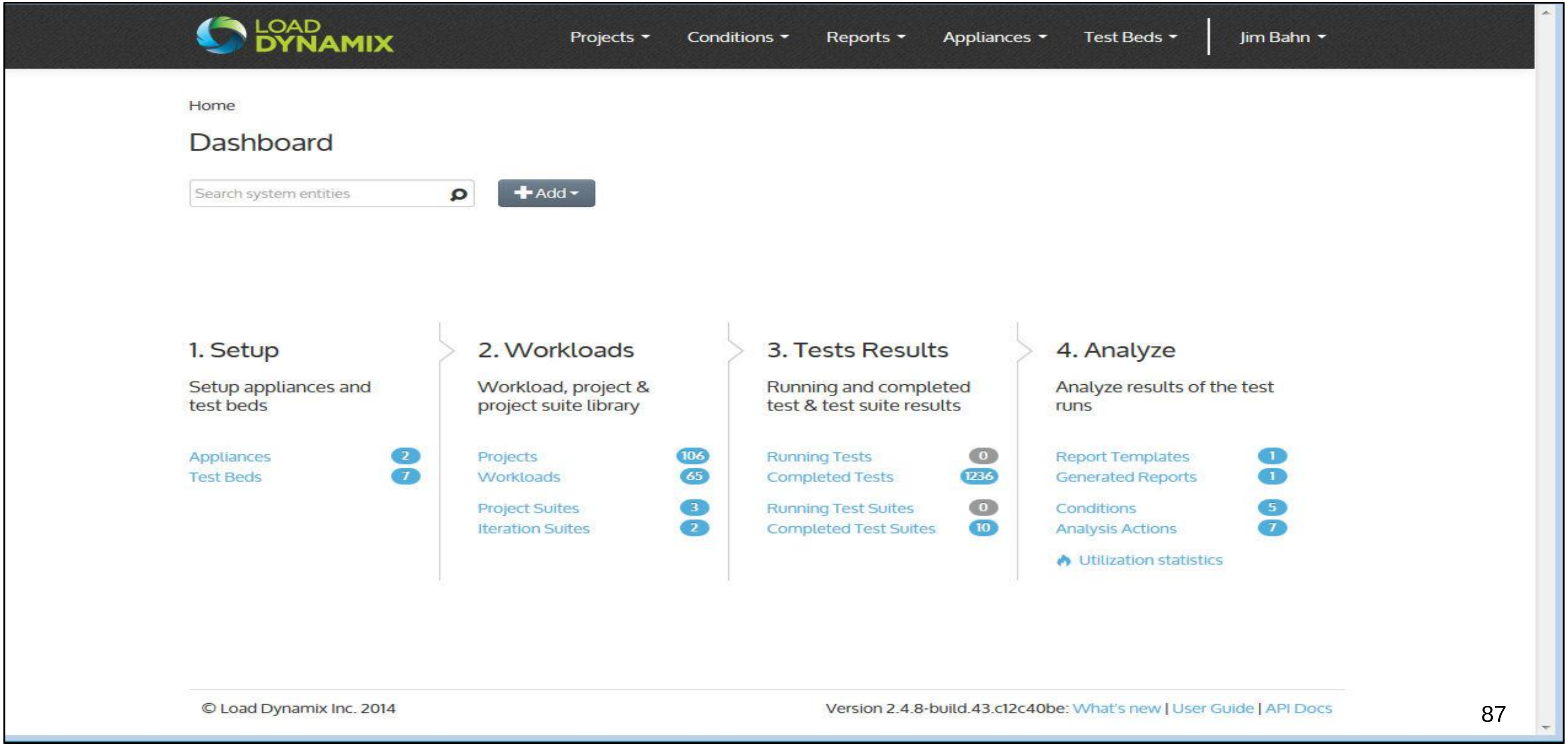
Name	Host IOs/sec	Read Response Time (ms)	Write Response Time (ms)	% Hit	% Writes	% Reads	% Read Miss	WP Count	Avg I/O Size	Capacity (GB)	% Used	%I/O Avg		Member	%RR	%SR	%RW	%SW
arc	2.8	1.7	7.7	100	98.4	1.6	2.9	0	347	256	98	0.10	2745.9	4			93	7
dbf1	522.1	2.7	0.8	51.5	0.4	99.6	48.7	10.9	19	256	95	19.01	2745.9	4	95	4		
dbf2	448.5	2.9	0.8	51.3	0.1	99.9	48.7	2.4	16	256	100	16.33	2745.9	4	94	6		
dbf3	316.6	1.8	1.2	82.5	5.2	94.8	18.4	96.8	19	256	100	11.53	2745.9	4	84	11	5	
dbf4	297	1.8	0.9	42.9	0.9	99.1	57.6	8	29	100	100	10.82	2745.9	2	99		1	
dbf5	235.6	1.4	1.2	87.2	4.8	95.2	13.4	65.8	17	256	100	8.58	2745.9	4	84	11	5	
dbf6	220.2	1.7	1	83.9	5.6	94.4	17	58.9	20	256	100	8.02	2745.9	4	84	11	5	
dbf7	201.4	3.3	1.4	82.7	1.5	98.5	17.6	17.2	237	256	95	7.33	2745.9	4	94	4	1	
dbf8	165.7	3.1	1.1	66.2	5.1	94.9	35.6	35.1	19	200	83	6.03	2745.9	4	91	3	5	
dbf9	91.9	1.3	2.2	88.3	6.2	93.8	12.4	24.7	17	100	100	3.35	2745.9	2	82	11	6	
dbf10	90.3	3.3	2.3	71.6	27.7	72.3	39.1	145.7	48	200	99	3.29	2745.9	4	73	1	26	
dbf11	7.6	5.4	1.3	57.9	17.8	82.2	51.3	6.3	105	256	100	0.28	2745.9	4	81	1	18	1
oraex	1.5	3.6	0.7	62.6	17.7	82.3	42.2	1.4	2	33	86	0.05	2745.9	1	82		17	2
quest	6.3	0.8	1.4	98.5	88.5	11.5	7.2	13.5	13	10	40	0.23	2745.9	1	9	2	86	4
redo1	70.2	6	0.7	87.9	96.9	3.1	20.3	63.6	28	32	93	2.56	2745.9	4	3		88	9
redo2	68.1	0.5	0.8	88	99.6	0.4	0.9	68.4	14	32	93	2.48	2745.9	4			90	9

I/O Metrics from Existing App on Storage Array

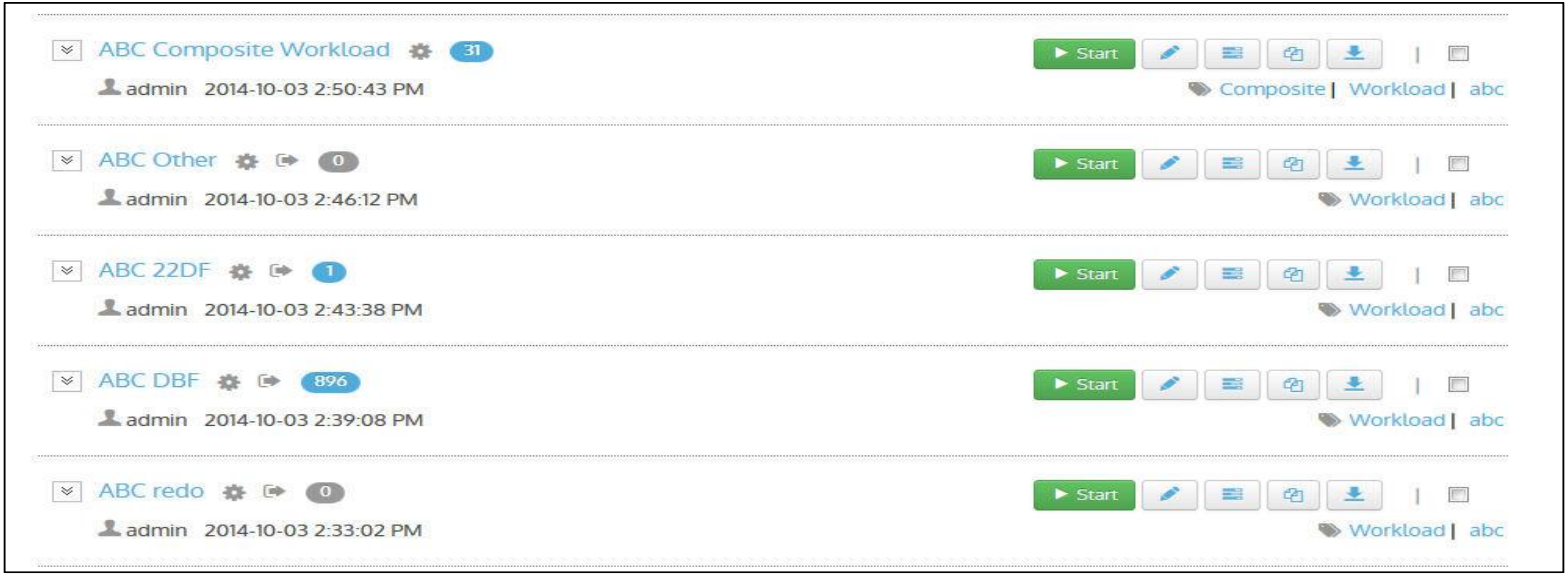
Sorted by common LUN I/O profiles

Name	Host IOs/sec	% Writes	%Reads	Avg I/O Size	Capacity (GB)	%RR	%SR	%RW	%SW
dbf1	522.1	0.4	99.6	19	256	95	4	0	0
dbf2	448.5	0.1	99.9	16	256	94	6	0	0
dbf3	316.6	5.2	94.8	19	256	84	11	5	0
dbf4	297	0.9	99.1	29	100	99	0	1	0
dbf5	235.6	4.8	95.2	17	256	84	11	5	0
dbf6	220.2	5.6	94.4	20	256	84	11	5	0
dbf7	165.7	5.1	94.9	19	200	91	3	5	0
dbf8	91.9	6.2	93.8	17	100	82	11	6	0
dbf9	90.3	27.7	72.3	48	200	73	1	26	0
dbf10	7.6	17.8	82.2	105	256	81	1	18	1
dbf11	201.4	1.5	98.5	237	256	94	4	1	0
redo1	70.2	96.9	3.1	28	32	3	0	88	9
redo2	68.1	99.6	0.4	14	32	0	0	90	9
quest	6.3	88.5	11.5	13	10	9	2	86	4
arc	2.8	98.4	1.6	347	256	0	0	93	7
oraex	1.5	17.7	82.3	2	33	82	0	17	2
dbf	**2395.5**	7.38	92.62	**30.90**	213.60	**87**	**6**	**7**	**0**
dbf11	**201.4**	1.5	98.5	**237**	256	**94**	**4**	**1**	**0**
redo	**138.3**	98.25	1.75	**21**	32	**2**	**0**	**89**	**9**
other	**10.6**	68.2	31.8	**120.7**	99.7	**30**	**1**	**65**	**4**

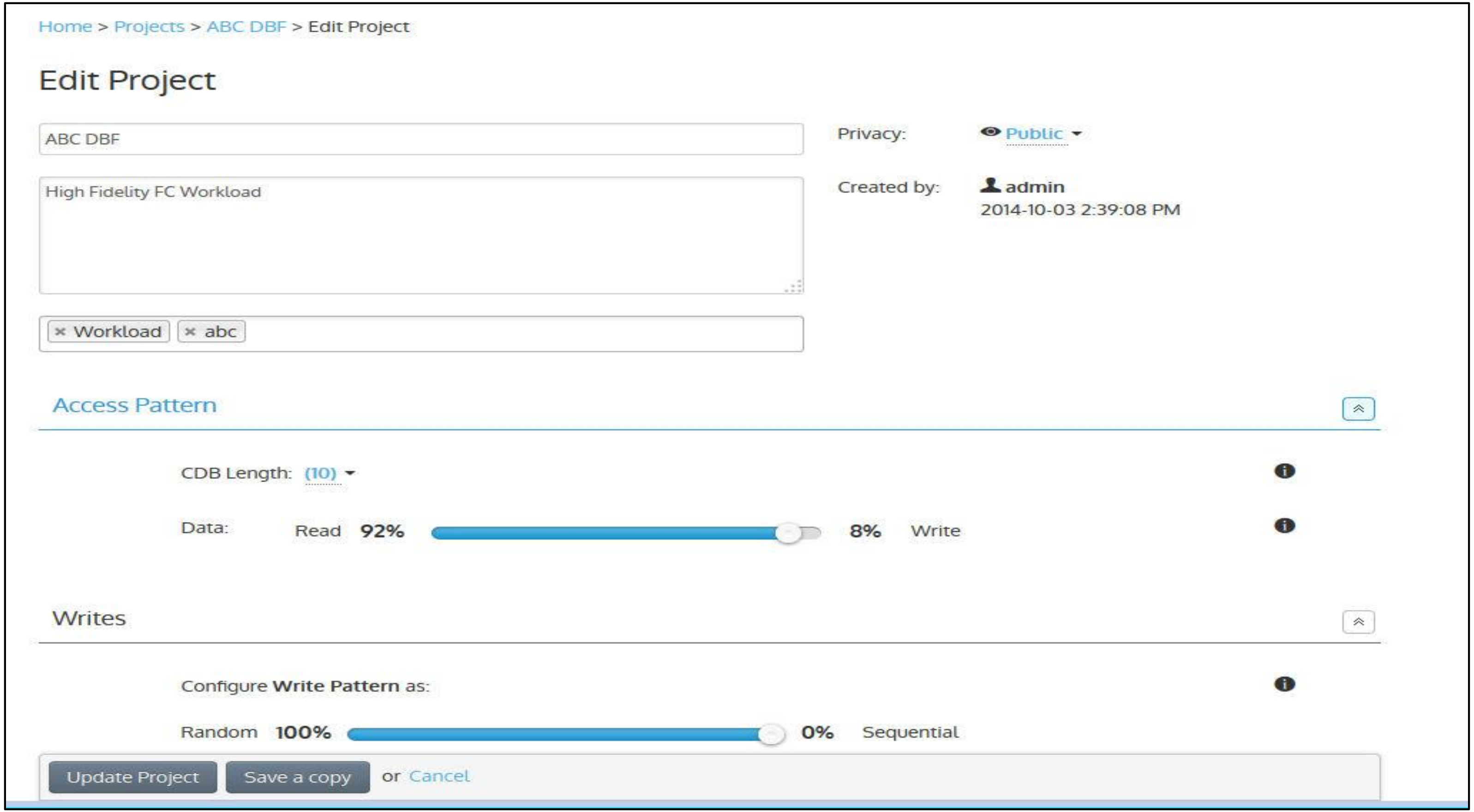
Load DynamiX Enterprise Dashboard



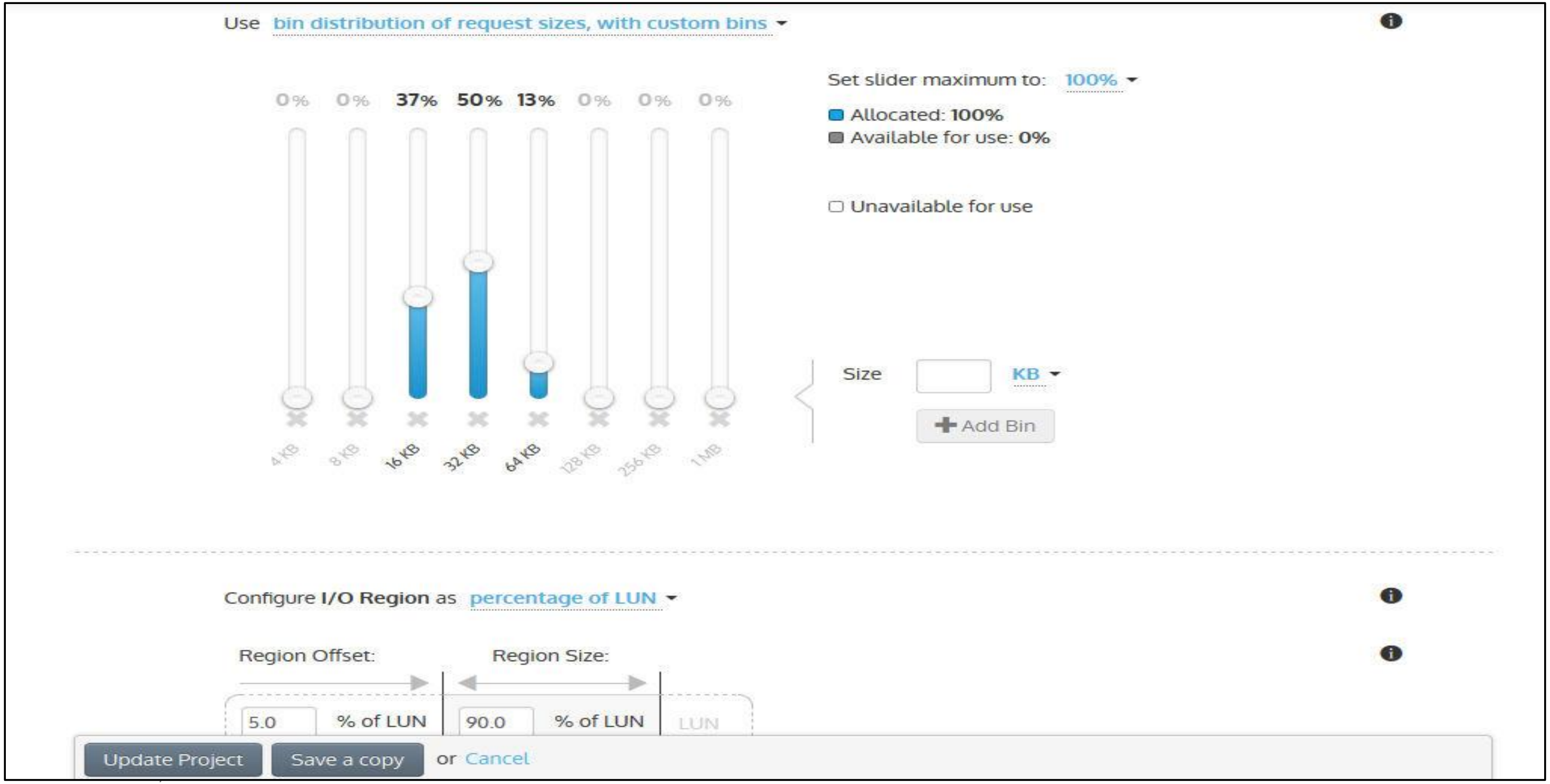
Create Workloads for Each LUN Group



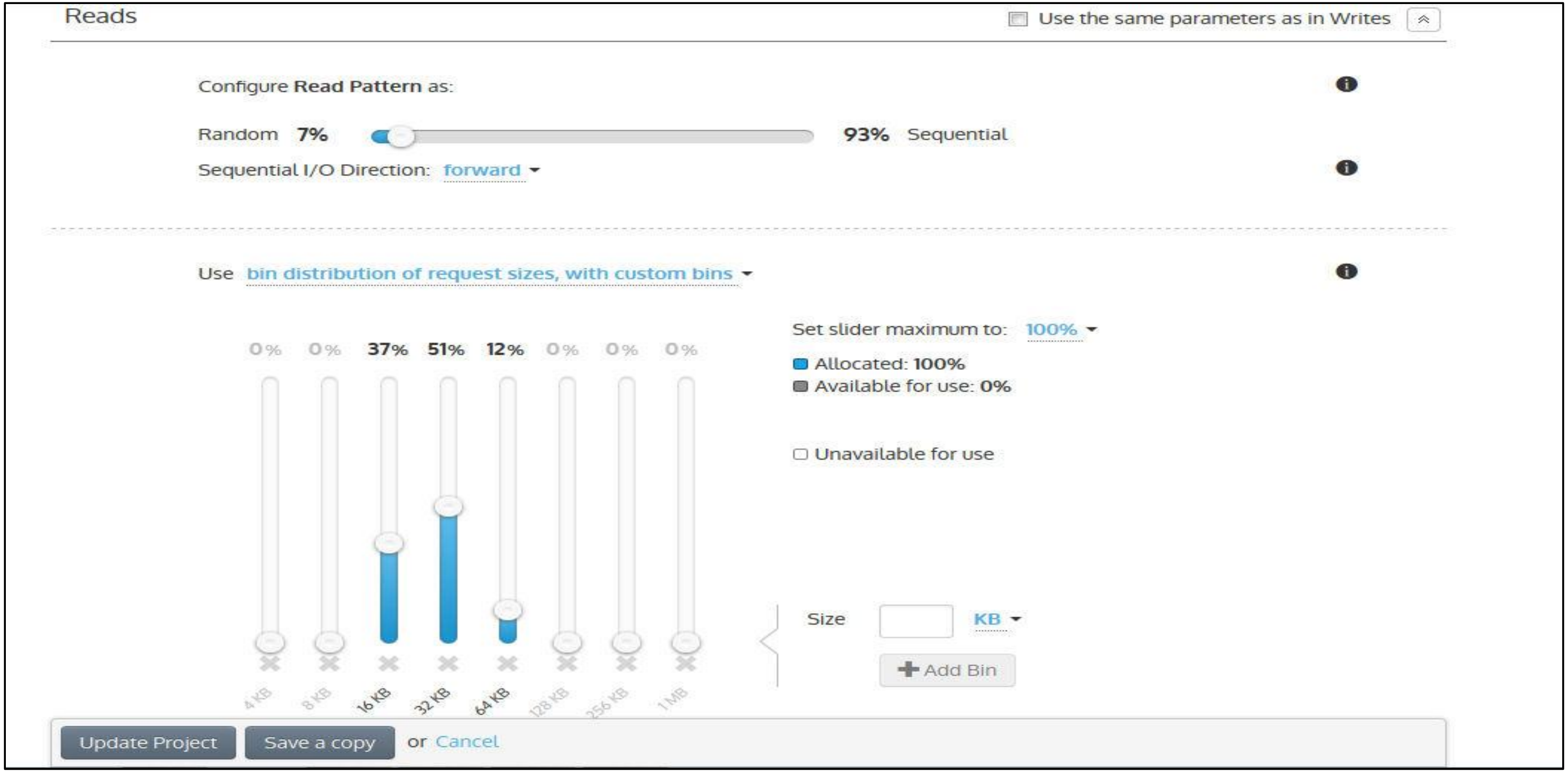
Build a Workload Model for Each LUN Group



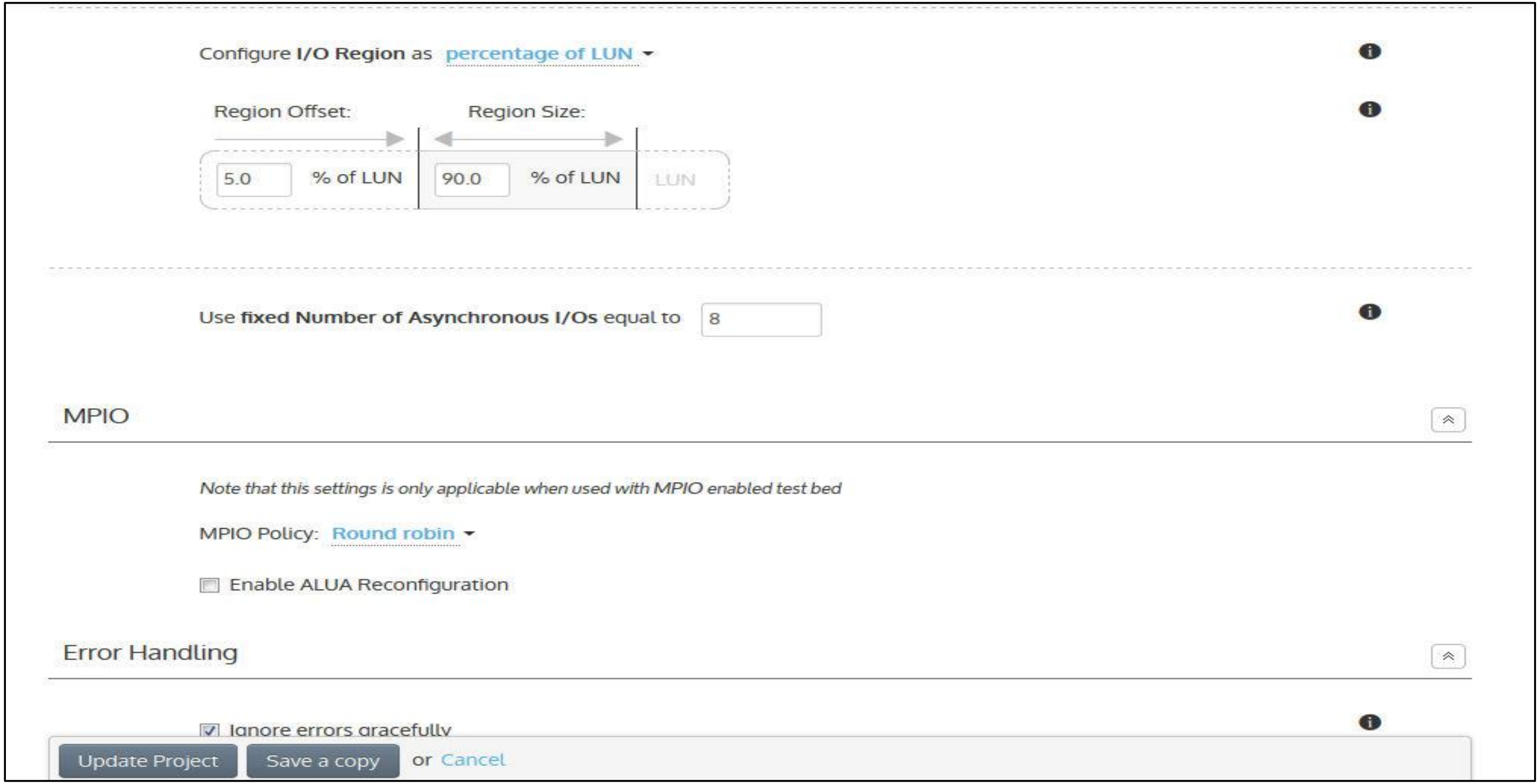
Build a Workload Model for Each LUN Group



Build a Workload Model for Each LUN Group



Build a Workload Model for Each LUN Group



Build a Workload Model for Each LUN Group

Data Parameters

Use random data content

Pre-test parameters

Specify how to run pre-test: Before the test

High Fidelity FC Workload pre-test

Configure **I/O Region** as percentage of LUN

Region Offset: 0.0 % of LUN

Region Size: 100.0 % of LUN

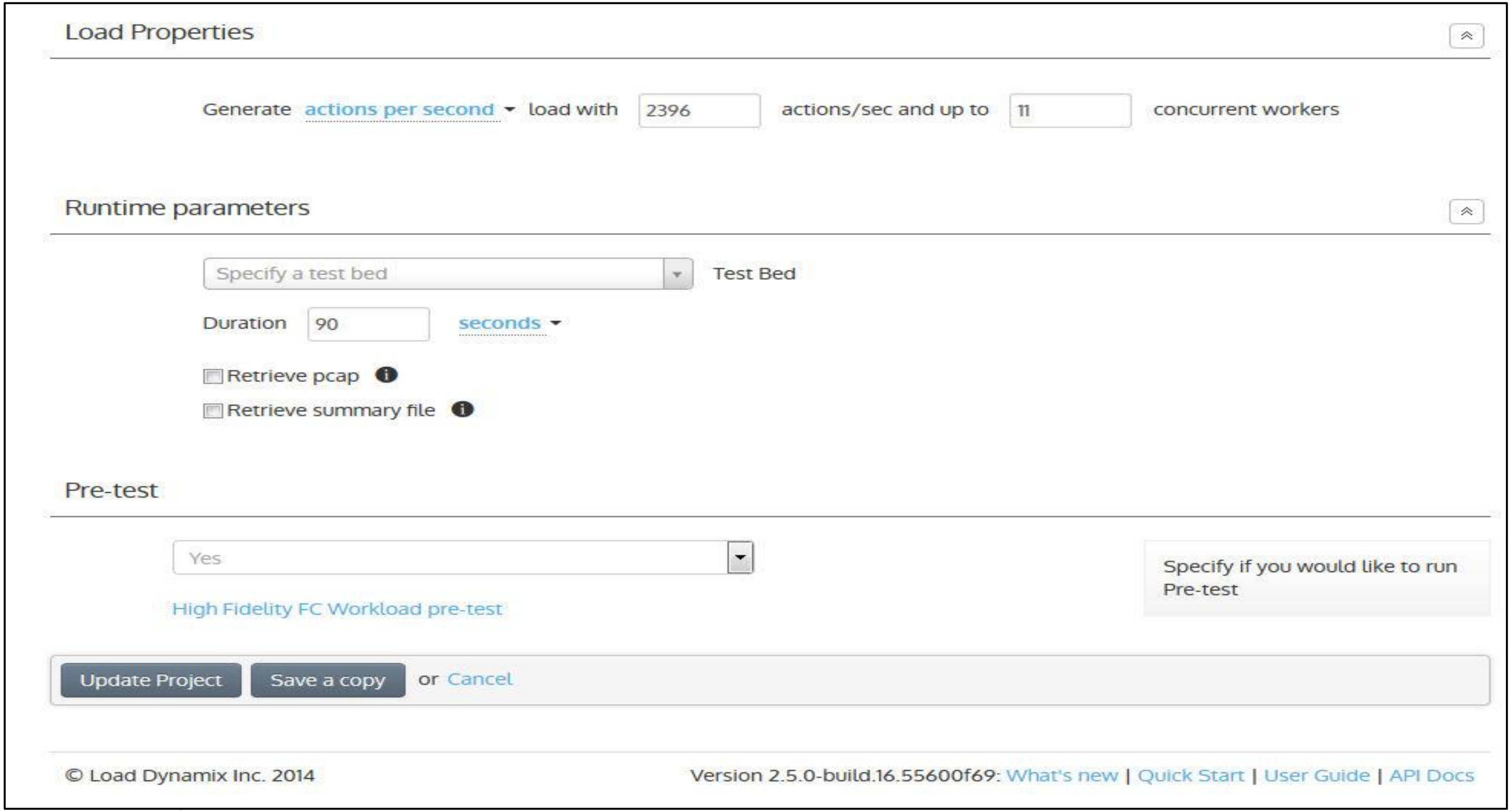
LUN

Repeats 1

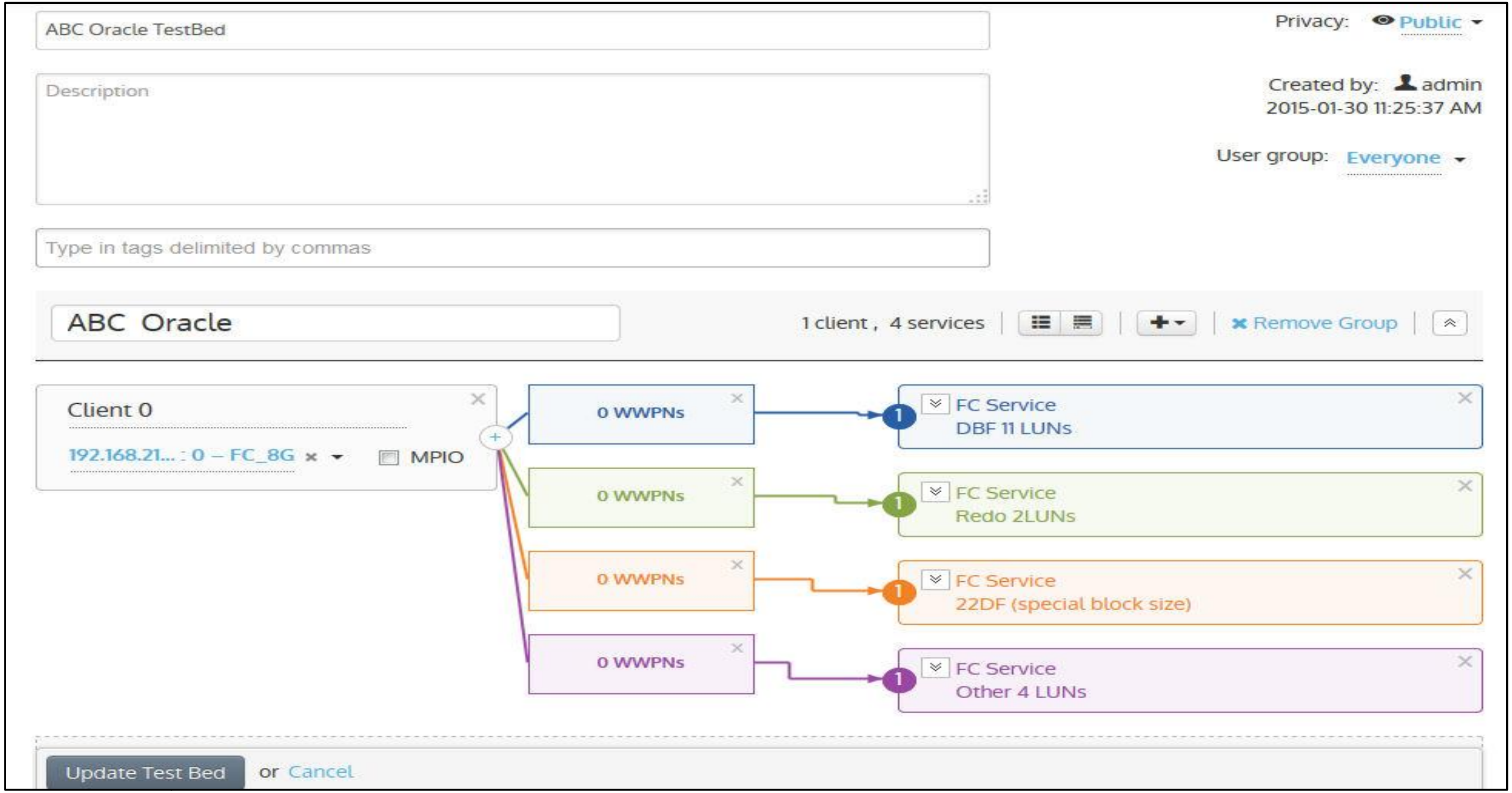
Load Properties

Update Project | Save a copy | or Cancel

Build a Workload Model for Each LUN Group

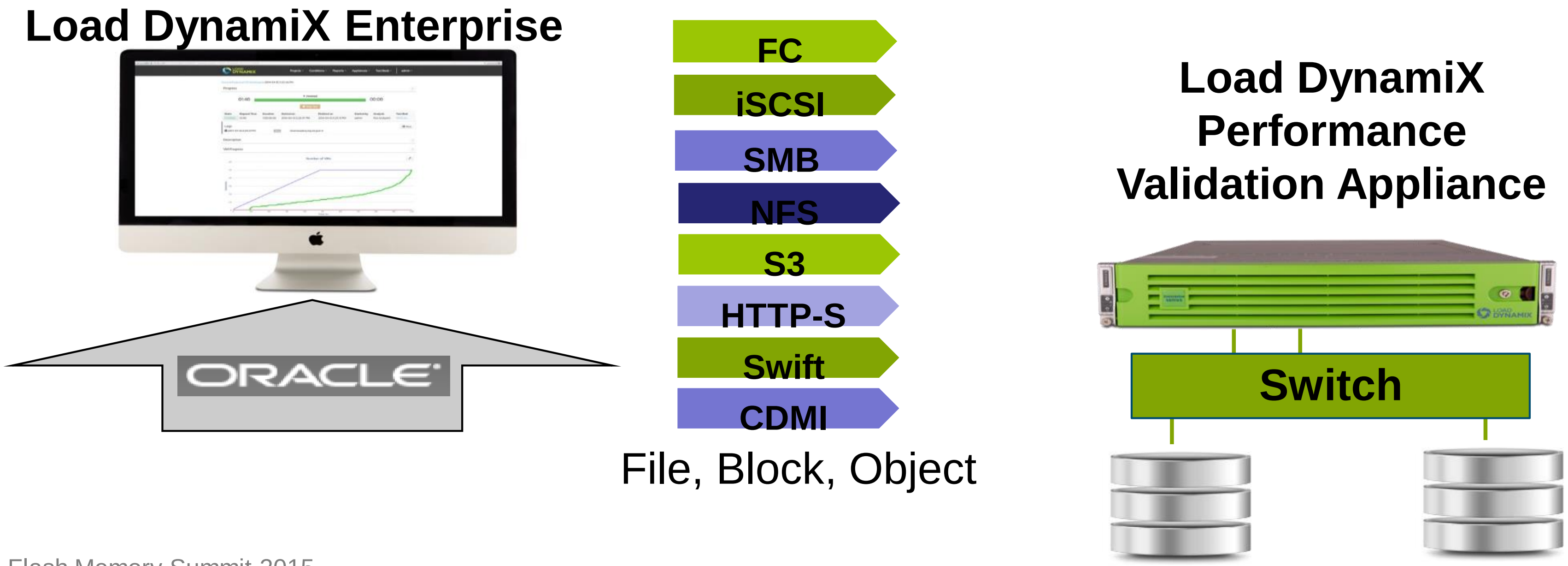


Build a Composite Workload on the Test Bed

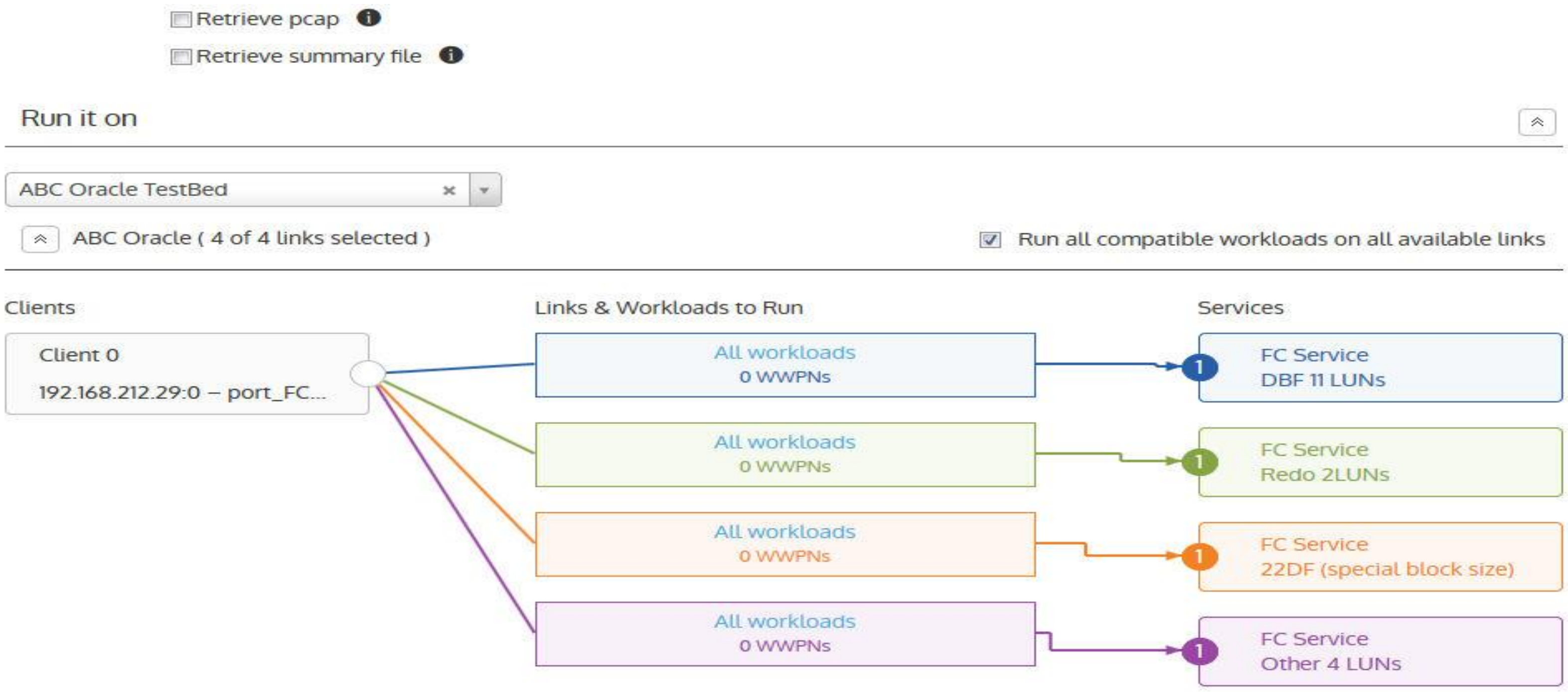


Methodology

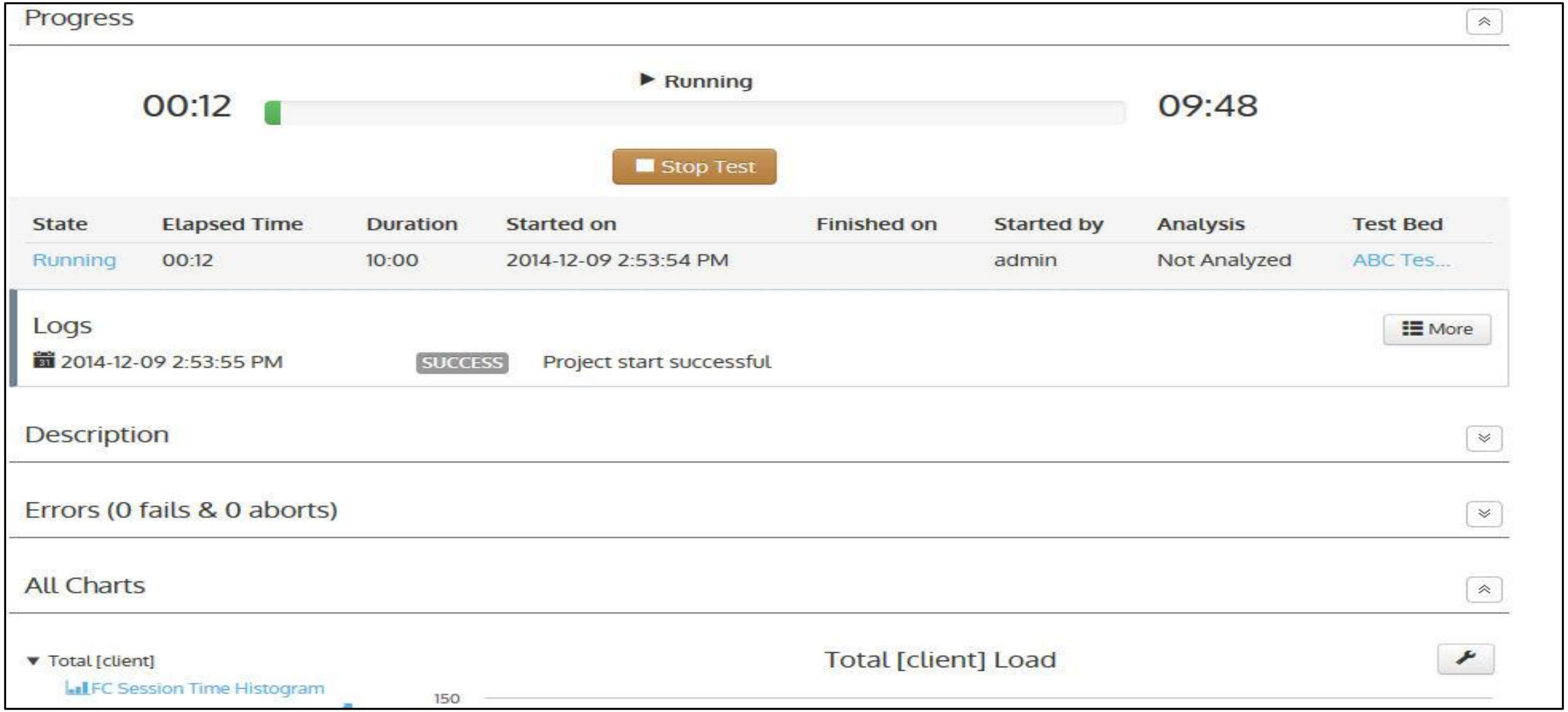
2 Deploy Test Configuration & Run Emulations



Run Composite Workload



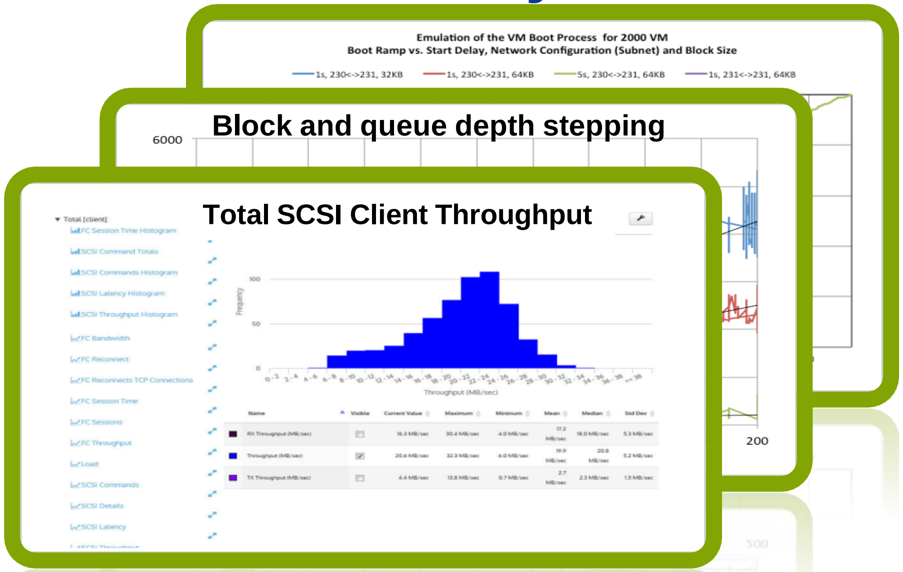
Run Composite Workload



Methodology

3 Analyze Results

Analytics



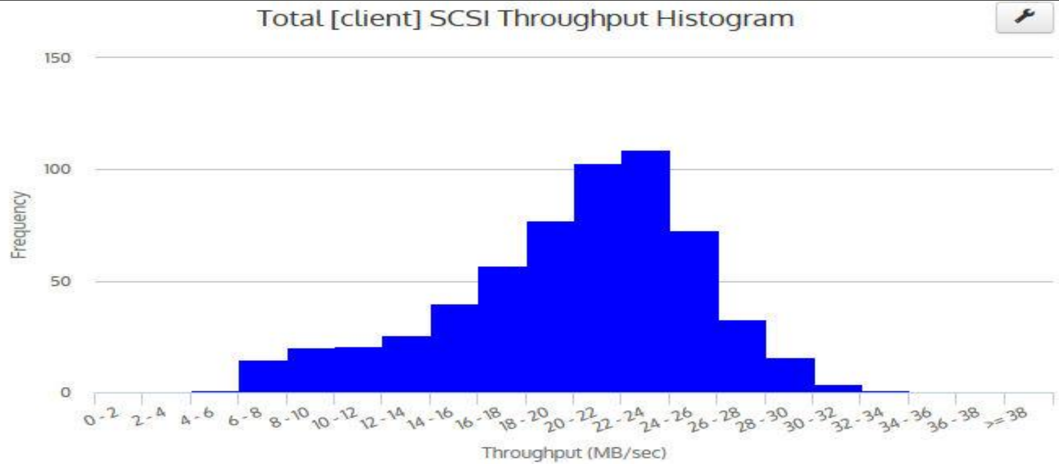
Insight

- Technology Evaluation
- Product Evaluation
- Configuration Optimization
- Pre-production staging validation
- Change validation

Analyze Report Results

Throughput histogram

- ▼ Total [client]
  - FC Session Time Histogram
  - SCSI Command Totals
  - SCSI Commands Histogram
  - SCSI Latency Histogram
  - SCSI Throughput Histogram
  - FC Bandwidth
  - FC Reconnect
  - FC Reconnects TCP Connections
  - FC Session Time
  - FC Sessions
  - FC Throughput
  - Load
  - SCSI Commands
  - SCSI Details
  - SCSI Latency
  - SCSI Throughput

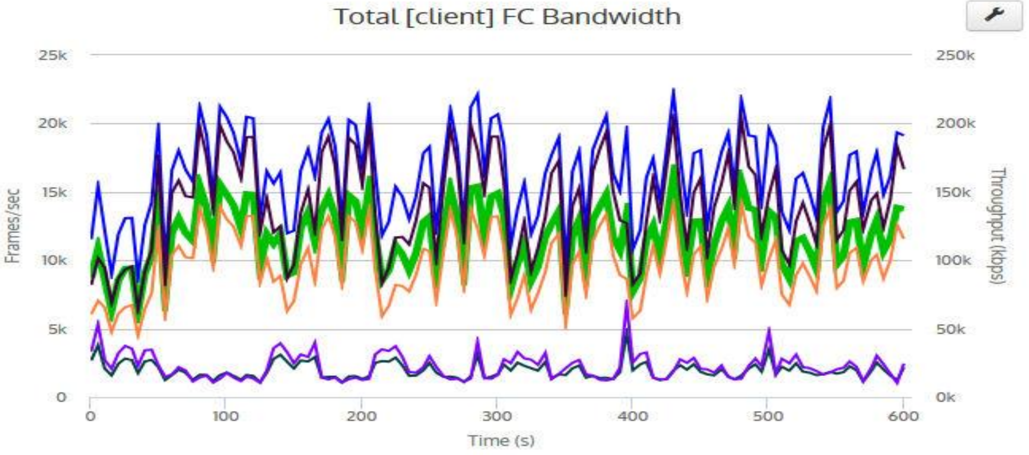


	Name	Visible	Current Value	Maximum	Minimum	Mean	Median	Std Dev
■	RX Throughput (MB/sec)	☐	16.3 MB/sec	30.4 MB/sec	4.0 MB/sec	17.2 MB/sec	18.0 MB/sec	5.3 MB/sec
■	Throughput (MB/sec)	☑	20.6 MB/sec	32.3 MB/sec	6.0 MB/sec	19.9 MB/sec	20.8 MB/sec	5.2 MB/sec
■	TX Throughput (MB/sec)	☐	4.4 MB/sec	13.8 MB/sec	0.7 MB/sec	2.7 MB/sec	2.3 MB/sec	1.5 MB/sec

Analyze Report Results

Bandwidth vs. Time

- Total [client]
  - FC Session Time Histogram
  - SCSI Command Totals
  - SCSI Commands Histogram
  - SCSI Latency Histogram
  - SCSI Throughput Histogram
  - FC Bandwidth
  - FC Reconnect
  - FC Reconnects TCP Connections
  - FC Session Time
  - FC Sessions
  - FC Throughput
  - Load
  - SCSI Commands
  - SCSI Details
  - SCSI Latency



Name	Visible	Anomalies	Trend Line	Current Value	Maximum	Minimum	Mean	Median	Std Dev
FC RX OK Frames/sec	☑	☐	☐	9,354	17,362.7	2,314	9,835.3	10,216	2,993.0
FC RX OK Throughput (kbps)	☑	☐	☐	133,432.8 kbps	249,262.2 kbps	32,435.2 kbps	141,499.5 kbps	147,613.3 kbps	43,319.8 kbps
FC TX OK Frames/sec	☑	☐	☐	2,888	7,530	666	1,951.8	1,793.2	813.1
FC TX OK Throughput (kbps)	☑	☐	☐	35,936 kbps	113,307.8 kbps	6,175.8 kbps	22,516.7 kbps	19,068.4 kbps	12,831.1 kbps
Frames/sec	☑	☐	☐	12,242	19,233.3	3,472	11,787.1	12,296	3,036.6
Throughput (kbps)	☑	☐	☐	169,368.8 kbps	264,844.7 kbps	49,138.6 kbps	164,016.2 kbps	170,910.8 kbps	42,010.3 kbps

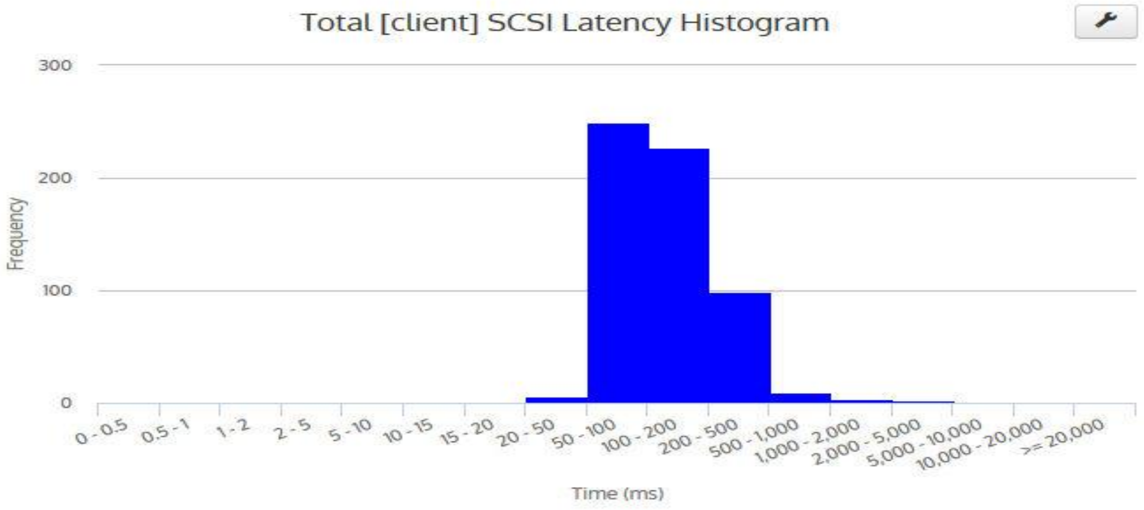
Analyze Report Results

Latency histogram

All Charts

▼ Total [client]

- FC Session Time Histogram
- SCSI Command Totals
- SCSI Commands Histogram
- SCSI Latency Histogram
- SCSI Throughput Histogram
- FC Bandwidth
- FC Reconnect
- FC Reconnects TCP Connections
- FC Session Time
- FC Sessions
- FC Throughput



Name	Visible	Anomalies	Current Value	Maximum	Minimum	Mean	Median	Std Dev
Average Response/Latency Time (ms)	☑	☐	3,555.1 ms	19,346.9 ms	43.2 ms	174.5 ms	110.7 ms	814.0 ms

Use Case Summary

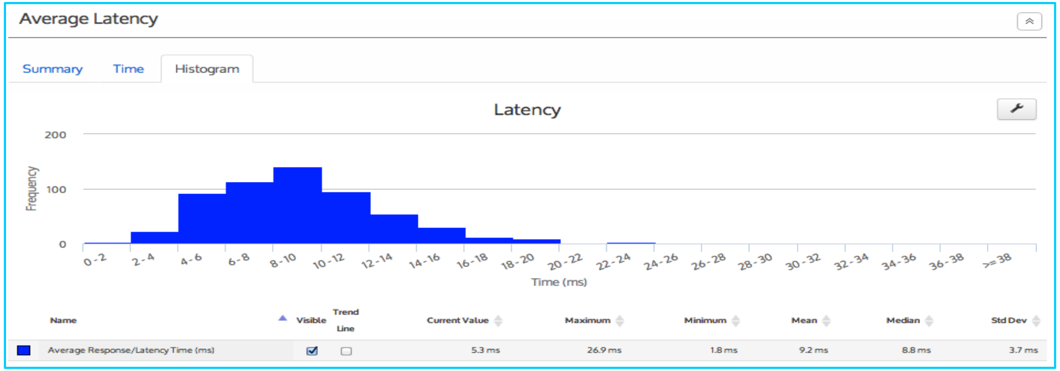
Test DB performance pre-deployment via workload modeling

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

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

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



Benefits of Using Load DynamiX

- **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 your Oracle application performance

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

Julia Palmer
Performance Engineering Manager

Learn More

Oracle Workload Modeling

- **Info sheet: 2 pages**
  http://www.loaddynamix.com/wp-content/uploads/2014/11/OracleWorkload_Infosheet.pdf

- **Video: 4 minute overview**
  http://www.loaddynamix.com/resources/videos-product/

Q & A