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

- HPC
- Data Warehouse
- Web Applications
- User Files
- Online Archive
- Transaction Processing
- VM Infrastructure

Greater Performance vs. More Capacity vs. Less Capacity
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
## Considerations for Solid State Storage Arrays

### Data Services Management

- **Data Reduction**
  - Deduplication
  - Compression
  - Thin Provisioning

- **Replication**
  - Local (writable)
  - Remote (Future)

- **Management**
  - Non-disruptive upgrades
  - REST APIs

### Investment Protection

- **Self-healing techniques** (Reliability)
- **Hardware Redundancy** (Availability)
- **Serviceability**

### Support

- **Hypervisor**
  - VMware vSphere
  - MS Hyper V

- **Scale out and Clustering**

- **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 Flash Arrays
- Wear Leveling
- Garbage collection
- Metadata Management
- Self-healing techniques
- Inline Data deduplication
- Inline Compression

### Traditional HDD Arrays
- 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

Repeatable non-compressible pattern

Repeatable non-compressible pattern

Repeatable non-compressible pattern
Traditional Disk Performance Curve

Ramp up

Steady State
Rotational Latency

Next Block To Read: Requires Full Rotation

Read Head Position
Stroke Latency

Next Block To Read: Read Head Must Move to New Track

Read Head Position
Worst-Case Latency

Next Block To Read: Read Head Must Move to New Track And Wait One Full Rotation

Read Head Position
SSD Performance States - Normalized IOPS

Normalized IOPS

Time (Minutes)

FOB

Pre-conditioning Transition

Write Cliff

Steady State (desirable test range)

(D1 MLC, D2 MLC, D3 MLC, D4 MLC, D5 MLC, D6 MLC, D7 SLC, D8 SLC)

(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
Pre-Conditioning

- Writing to every cell to achieve steady state
- Helps to ensure garbage collection during main test cycles

Involves breaking in entire flash array

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

19/8/2015 Flash Memory Summit All Rights Reserved
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 capacity

- Magnifying the load to test future expected loads
Methodology Components
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

Repeatable non-compressible pattern

Repeatable non-compressible pattern

Repeatable non-compressible pattern
## Thread Count and Queue Depth

<table>
<thead>
<tr>
<th>Helps find max performance for each:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Thread count</td>
</tr>
<tr>
<td>• Queue depth</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tests must find max IOPs an array can do per:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Thread count (workers)</td>
</tr>
<tr>
<td>• Queue depth (outstanding I/Os)</td>
</tr>
<tr>
<td>• Combination of threads and queue depth</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Increasing thread count</th>
</tr>
</thead>
<tbody>
<tr>
<td>• past current requirements shows how array meets future needs</td>
</tr>
</tbody>
</table>
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
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
- Metrics
- Workload

Creating a Model
Workload Preconditioning

- Per-LUN sequentiality vs. randomness
- I/O size distribution
- Per LUN I/O writes rate
- Compressibility and Duplicability

Elements in a preconditioning workload
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 test for 10 hours
- Measure whether steady state is achieved
- Use steady state as defined in PTS spec

Run workload-dependent data stream to prepare data set

- 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

Measure how an array performs as the percentage of the array is filled?

- Thin provisioning is turned on if available
- Deduplication/Compression
- Capture at various IO rates and measure variation

Ensure enterprise features are turned on
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

Flash Storage Testing Using Vdbench

Dennis Martin
Flash Storage Testing Using Vdbench
Pre-conference session

Dennis Martin, President
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 testing
- Website: www.demartek.com/TestLab
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

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
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
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
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
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.
The nature of each workload has a large impact on latency.
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
• 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
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

Demartek

Dell PowerEdge R820

LoadDynamIX Appliance

Nimbus Data S-Class All-flash Array
Thank You!

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

Performance Analytics

ORACLE
Workload Modeling & Performance Profiling

Load DynamiX Performance Validation Appliance

Switch
Why IT Organizations Use Load DynamiX

Use cases

- Technology Evaluation
  Flash, NFSv4, FC, OpenStack, Ceph?

- Change Validation
  Effect of HW and SW changes?

- Pre-Production Staging Validation
  Hot staging and burn-in?

- Product Evaluation
  Best product for your workloads?

- Configuration Optimization
  Media, tiering, etc?
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)

*DB apps do not present a single I/O profile
## I/O Metrics via Storage Monitoring Tool

Via existing app on storage array

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

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

Flash Memory Summit
Santa Clara, CA
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

Configure Read Pattern as:
- Random: 7%
- Sequential: 93%
- Sequential I/O Direction: forward

Use bin distribution of request sizes, with custom bins:

Set slider maximum to: 100%
- Allocated: 100%
- Available for use: 0%

Size: KB

Update Project, Save a copy, or Cancel
Build a Workload Model for Each LUN Group

Configure I/O Region as \textit{percentage of LUN}.

- Region Offset: \(5.0\%\) of LUN
- Region Size: \(90.0\%\) of LUN

Use fixed Number of Asynchronous I/Os equal to 8.

\textbf{MPIO}

\textit{Note that this settings is only applicable when used with MPIO enabled test bed.}

- MPIO Policy: \textit{Round robin}

- \(\square\) Enable ALUA Reconfiguration

\textbf{Error Handling}

- \(\Box\) Ignore errors gracefully

\textbf{Update Project} \textbf{Save a copy} or \textbf{Cancel}
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

Configure I/O Region as percentage of LUN

Region Offset:

<table>
<thead>
<tr>
<th>Offset</th>
<th>% of LUN</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

Region Size:

<table>
<thead>
<tr>
<th>Size</th>
<th>% of LUN</th>
</tr>
</thead>
</table>

Repeats: 1
Build a Workload Model for Each LUN Group

Load Properties

Generate actions per second load with 2396 actions/sec and up to 11 concurrent workers

Runtime parameters

Specify a test bed

Duration 90 seconds

Retrieve pcap

Retrieve summary file

Pre-test

Yes

High Fidelity FC Workload pre-test

Update Project  Save a copy or Cancel
Build a Composite Workload on the Test Bed

ABC Oracle TestBed

Description:

Type in tags delimited by commas

ABC Oracle

1 client, 4 services

192.168.21... 0 - FC_8G

0 WWPNs

FC Service
DBF 11 LUNs

FC Service
Redo 2LUNs

FC Service
22DF (special block size)

FC Service
Other 4 LUNs

Update Test Bed or Cancel
Methodology

2. Deploy Test Configuration & Run Emulations

Load DynamiX Enterprise

- FC
- iSCSI
- SMB
- NFS
- S3
- HTTP-S
- Swift
- CDMI

File, Block, Object

Load DynamiX Performance Validation Appliance

Switch
Run Composite Workload
Run Composite Workload
Analyze Results

Analytics

Methodology

Insight

- Technology Evaluation
- Product Evaluation
- Configuration Optimization
- Pre-production staging validation
- Change validation
Analyze Report Results

Throughput histogram
Analyze Report Results

Bandwidth vs. Time
Analyze Report Results
Latency histogram
1. Characterize composite workload from existing metrics (and other figures) as related to the mission critical database workload captured from the existing (old) array.

2. Model profile 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.
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

---

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**

- **Video: 4 minute overview**