Utilizing VDBench to Perform IDC AFA Testing

Michael Ault, IBM
Oracle FlashSystem Consulting Manager
The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.

Member companies and individual members may use this material in presentations and literature under the following conditions:

- Any slide or slides used must be reproduced in their entirety without modification.
- The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.

This presentation is a project of the SNIA Education Committee.

Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.

The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.
Abstract

Utilizing VDBench to Perform IDC AFA Testing

- This session will review what an all-flash-array consists of, what IDC is and what their testing involves and how to use VDBench to perform AFA testing and comparisons.
Agenda

- What is an AFA?
- What is VDBench?
- What is the IDC and why do we care?
- What are the suggested AFA tests?
- Implementing the IDC AFA tests with VDBench
What is an AFA?

- All-Flash-Array
- Not SSD (form-factor flash drives)
- Built from Function-designed flash modules
- Integrated control, configuration and monitoring
- S4-TWG defines AFA as a solid state storage array connected by redundant network protocols
What is VDBench?

- Command line system
- Simulates various load scenarios including multiple hosts
- Doesn’t require an Oracle/DB install
- Scriptable
- Provided free from Oracle
- Vdbench has been tested on Solaris Sparc and x86, Windows NT, 2000, 2003, 2008, 2012, XP and Windows 7+8, HP/UX, AIX, Linux, Mac OS X, zLinux, and native VmWare
- Download from:
VDBench Download Page

VDBench is a command line utility specifically created to help engineers and customers generate disk I/O workloads to be used for validating storage performance and storage data integrity. VDBench execution parameters may also be specified via an input text file.

You must accept the OTN License Agreement to download this software.

Accept License Agreement | Decline License Agreement

Vdbench (5.04.02)
- vdbench-50402.zip
  - User Guide
  - Vdbench Discussion Forum

Vdbench (5.04)
- vdbench-50401.zip
  - User Guide
  - Vdbench Discussion Forum

Related Downloads
- Vdbench source code
IDC and AFA

- IDC provides guidelines on standard tests to use in comparing computer technologies
- IDC has provided a detailed test guidelines for AFA
- Using the IDC guidelines you can easily test and compare AFAs
- The public link for the IDC testing guidelines for AFAs is: http://idcdocserv.com/251951
Differences Between AFA and HDD

- **Preconditioning.**
  - Flash-based array performance will differ significantly from FOB (fresh out of box) and after first write on every flash cell.
  - HDDs don’t exhibit the FOB effect.

- **Read/write asymmetries.**
  - Traditional HDD overwrites data in place and does not lock data on writes.
  - Flash needs to erase before rewriting and cell locks on writes causing writes to be slower.

- **Endurance.**
  - After a defined number of P/E cycles, flash will become read only and ultimately inoperable.
Data stream/data set definition.

- The mixed virtual workloads in 3rd Platform computing are very different than traditional client/server workloads.
- A relevant workload will exhibit
  - a variety of read/write ratios
  - wide distribution of block sizes
  - Be skewed toward random I/O
  - Have a high percentage of reducible data

Load generation.

- Flash-based arrays deliver 10x performance
- Load generators have to take this into account
Some Flash Terminology

❖ P/E cycles.
  ◆ Each individual flash cell P/E cycle, the flash cell incurs a slight amount of damage known as wear.
  ◆ A single iteration of erasing and writing to a flash cell is known as a "program/erase cycle" (P/E cycle).

❖ Asymmetric behavior.
  ◆ flash delivers latencies an order of magnitude lower than HDDs,
  ◆ read/write I/O exhibit different latencies depending on state of flash media
  ◆ If the page has data that must be overwritten, a P/E cycle must occur
Some Flash Terminology

- Overprovisioned capacity.
  - A set amount of capacity reserved for administrative activities associated with flash performance, endurance, or reliability.

- Wear leveling.
  - Cells have a specified wear rating.
  - Frequent use cells will wear out sooner
Some Flash Terminology

Free space management.
- To write a single byte, an entire erase block must be available.
- "garbage collection," ensures that systems always have a ready supply of clean blocks.
- Invalidated data is not immediately overwritten but marked as "available" to be written.

Write amplification.
- There are two processes in most flash-based arrays that amplify the number of writes within the array: garbage collection and data protection.
- The ratio of FTL writes to Server requests
Some Flash Terminology

❖ **Flash translation layer (FTL).**
  - Flash storage is generally combined with an FTL (FPGA, ASIC, or software based)
  - FTL is a logical block interface for OS to access the flash media as if it were a traditional HDD.
IDC Testing

In the universe of performance testing, there are five basic types of tests that could potentially be performed, and these are ranked in order of relevance to understanding how an array will perform on your actual production workload:

- **Class A**: Workload testing using your actual data streams, data sets, and workflows
- **Class B**: Workload testing using data streams, data sets, and workflows that are generated using application-specific testing tools like SLOB (Oracle), Jetstress (Exchange), Login VSI (VDI), and others
- **Class C**: Workload testing that uses data streams, data sets, and workflows generated by general-purpose workload generation tools like Load DynamiX, FIO, or vdbench that are capable of very accurate modeling
- **Class D**: An intelligent performance corners test using generic data sets, data streams, and workflows designed to closely model 3rd Platform computing workloads
- **Class E**: Hero testing that encompasses most classic performance corners testing

In AFA testing, IDC recommends Class D type tests
# IDC Test Plan

## AFA Test Plan Summary

<table>
<thead>
<tr>
<th>Phase</th>
<th>Description</th>
<th>Tasks</th>
<th>Estimated Elapsed Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Setup and preconditioning</td>
<td>2 x 85% overwrite</td>
<td>24 hours</td>
</tr>
<tr>
<td></td>
<td>Data set preparation</td>
<td>2 x 5% overwrite</td>
<td>6 hours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 x 1% overwrite</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Baseline workload testing</td>
<td>Develop scripts</td>
<td>6 hours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thread/queue depth optimization</td>
<td>6 hours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IOPS ramp testing (4 runs at 3 hours per run)</td>
<td>12 hours</td>
</tr>
<tr>
<td>3</td>
<td>Functional testing</td>
<td>Document workflows</td>
<td>6 hours</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test workflows (4 runs at 4 hours per run)</td>
<td>16 hours</td>
</tr>
<tr>
<td>4</td>
<td>Fault injection testing</td>
<td>Inject failures (4 runs at 2 hours per run)</td>
<td>8 hours</td>
</tr>
<tr>
<td>5</td>
<td>Soak test</td>
<td>Run 4 baseline workloads consecutively (12 hours each)</td>
<td>48 hours</td>
</tr>
</tbody>
</table>

Source: IDC, 2014
Testing the a 40 TiB System

**Preconditioning**

- Actual amount of flash is 65.9 TB
- System 40 usable TiB (37.5 TB after RAID5)
- We already have extracted the “overhead” so we will configure the 37 into 37 -1 TiB LUNS
- Preconditioning volume of writes would be 65.9*0.85*2=112 TB
- At 128K blocks the throughput for the 840 is 1.7 GB/s for a single Power8 server with 16 threads per LUN (592 total)
- 67,500 seconds to do writes (~19 hours)
- Monitor for write cliff, if not reached at end of first run, repeat
Testing the a 40 TiB System

Data Set Preparation and Aging

- 2 additional runs with small random writes (4-16kb) each at 5% of 65.9 TB so 3.3 TB each run randomly spread across system (3 runs of around 1,100 GiB one at 4k, 8k and 16k)
- 5 additional runs with small random writes (4-16kb) each at 1% of 65.9 TiB (660 GB) each run randomly spread across system (3 streams 4, 8 and 16k each doing 220 GB each) each run addressing different sets of luns.
VDBench Scripting

- Utilizes tags for each script section
- SD for drive/storage designations
- WD for workload definitions
- RD for run definitions
Preconditioning Script

Create several luns, do 37-1.0 TiB size that used most of the 37.5 TiB.

Contents of precondition.cfg:

```
sd=default, openflags=o_direct
sd=sd1,lun=\\PhysicalDrive1
sd=sd2,lun=\\PhysicalDrive2
sd=sd3,lun=\\PhysicalDrive3
...
sd=sd37,lun=\\PhysicalDrive37

wd=fill,sd=sd*, rdpct=0, xfersize=128k, seekpct=-1
rd=precondition, wd=fill, iorate=max, elapsed=67500, interval=30,
threads=16
```
Running the Script

- Use the command line
- Specify output location if you want a specific location for the output HTML files

```
vdbench\vdbench -f precondition.cfg –m 37 –o \vdbench\output\precon
```
Example Output Files
## File Contents (Summary file)

<table>
<thead>
<tr>
<th>Sep 16, 2014</th>
<th>interval</th>
<th>i/o rate</th>
<th>MB/sec 1024**2</th>
<th>bytes</th>
<th>read</th>
<th>resp</th>
<th>read</th>
<th>write</th>
<th>resp</th>
<th>resp max</th>
<th>stddev</th>
<th>depth</th>
<th>sys%</th>
<th>cpu%</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:19:08.058</td>
<td>1</td>
<td>74950.00</td>
<td>36.60</td>
<td>512</td>
<td>10.04</td>
<td>0.341</td>
<td>0.541</td>
<td>0.318</td>
<td>426.042</td>
<td>1.776</td>
<td>25.5</td>
<td>0.0</td>
<td>4.3</td>
<td></td>
</tr>
<tr>
<td>16:19:09.018</td>
<td>2</td>
<td>103950.00</td>
<td>50.76</td>
<td>512</td>
<td>10.01</td>
<td>0.258</td>
<td>0.437</td>
<td>0.238</td>
<td>6.256</td>
<td>0.257</td>
<td>26.8</td>
<td>8.2</td>
<td>6.4</td>
<td></td>
</tr>
<tr>
<td>16:19:10.016</td>
<td>3</td>
<td>118267.00</td>
<td>57.75</td>
<td>512</td>
<td>10.13</td>
<td>0.233</td>
<td>0.464</td>
<td>0.207</td>
<td>46.793</td>
<td>0.540</td>
<td>27.5</td>
<td>13.6</td>
<td>7.6</td>
<td></td>
</tr>
<tr>
<td>16:19:11.030</td>
<td>4</td>
<td>121366.00</td>
<td>59.26</td>
<td>512</td>
<td>9.88</td>
<td>0.230</td>
<td>0.471</td>
<td>0.204</td>
<td>195.091</td>
<td>0.932</td>
<td>27.9</td>
<td>12.1</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td>16:19:12.020</td>
<td>5</td>
<td>110907.00</td>
<td>54.15</td>
<td>512</td>
<td>9.93</td>
<td>0.245</td>
<td>0.425</td>
<td>0.225</td>
<td>6.435</td>
<td>0.258</td>
<td>27.2</td>
<td>5.1</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>16:19:13.017</td>
<td>6</td>
<td>105364.00</td>
<td>51.45</td>
<td>512</td>
<td>9.99</td>
<td>0.261</td>
<td>0.427</td>
<td>0.243</td>
<td>7.380</td>
<td>0.248</td>
<td>27.5</td>
<td>8.3</td>
<td>5.4</td>
<td></td>
</tr>
<tr>
<td>16:19:14.016</td>
<td>7</td>
<td>125217.00</td>
<td>61.14</td>
<td>512</td>
<td>9.95</td>
<td>0.220</td>
<td>0.474</td>
<td>0.192</td>
<td>29.523</td>
<td>0.637</td>
<td>27.5</td>
<td>10.2</td>
<td>8.2</td>
<td></td>
</tr>
<tr>
<td>16:19:15.030</td>
<td>8</td>
<td>116730.00</td>
<td>57.00</td>
<td>512</td>
<td>10.10</td>
<td>0.239</td>
<td>0.486</td>
<td>0.212</td>
<td>149.333</td>
<td>0.830</td>
<td>27.9</td>
<td>12.5</td>
<td>7.8</td>
<td></td>
</tr>
<tr>
<td>16:19:16.018</td>
<td>9</td>
<td>117267.00</td>
<td>57.26</td>
<td>512</td>
<td>10.09</td>
<td>0.232</td>
<td>0.423</td>
<td>0.210</td>
<td>11.671</td>
<td>0.265</td>
<td>27.2</td>
<td>8.0</td>
<td>7.1</td>
<td></td>
</tr>
<tr>
<td>16:19:17.016</td>
<td>10</td>
<td>126531.00</td>
<td>61.78</td>
<td>512</td>
<td>10.01</td>
<td>0.218</td>
<td>0.462</td>
<td>0.190</td>
<td>16.961</td>
<td>0.605</td>
<td>27.5</td>
<td>10.4</td>
<td>7.5</td>
<td></td>
</tr>
<tr>
<td>16:19:18.002</td>
<td>11</td>
<td>111330.00</td>
<td>54.36</td>
<td>512</td>
<td>10.04</td>
<td>0.247</td>
<td>0.426</td>
<td>0.227</td>
<td>11.312</td>
<td>0.261</td>
<td>27.5</td>
<td>9.5</td>
<td>7.5</td>
<td></td>
</tr>
<tr>
<td>16:19:19.016</td>
<td>12</td>
<td>125546.00</td>
<td>61.30</td>
<td>512</td>
<td>10.15</td>
<td>0.219</td>
<td>0.471</td>
<td>0.191</td>
<td>62.544</td>
<td>0.619</td>
<td>27.5</td>
<td>12.8</td>
<td>8.6</td>
<td></td>
</tr>
<tr>
<td>16:19:20.017</td>
<td>13</td>
<td>108123.00</td>
<td>52.79</td>
<td>512</td>
<td>10.06</td>
<td>0.255</td>
<td>0.437</td>
<td>0.234</td>
<td>13.263</td>
<td>0.272</td>
<td>27.5</td>
<td>6.6</td>
<td>4.7</td>
<td></td>
</tr>
<tr>
<td>16:19:21.016</td>
<td>14</td>
<td>118485.00</td>
<td>57.85</td>
<td>512</td>
<td>10.03</td>
<td>0.233</td>
<td>0.459</td>
<td>0.207</td>
<td>16.959</td>
<td>0.565</td>
<td>27.6</td>
<td>10.2</td>
<td>7.4</td>
<td></td>
</tr>
<tr>
<td>16:19:22.030</td>
<td>15</td>
<td>127885.00</td>
<td>62.44</td>
<td>512</td>
<td>9.98</td>
<td>0.218</td>
<td>0.460</td>
<td>0.192</td>
<td>31.209</td>
<td>0.660</td>
<td>27.9</td>
<td>12.0</td>
<td>9.9</td>
<td></td>
</tr>
<tr>
<td>16:19:23.018</td>
<td>16</td>
<td>108483.00</td>
<td>52.97</td>
<td>512</td>
<td>9.99</td>
<td>0.251</td>
<td>0.449</td>
<td>0.229</td>
<td>301.127</td>
<td>0.958</td>
<td>27.2</td>
<td>7.7</td>
<td>5.8</td>
<td></td>
</tr>
<tr>
<td>16:19:24.016</td>
<td>17</td>
<td>108807.00</td>
<td>53.13</td>
<td>512</td>
<td>9.77</td>
<td>0.253</td>
<td>0.494</td>
<td>0.227</td>
<td>78.131</td>
<td>0.903</td>
<td>27.6</td>
<td>10.8</td>
<td>6.3</td>
<td></td>
</tr>
<tr>
<td>16:19:25.018</td>
<td>18</td>
<td>114688.00</td>
<td>56.00</td>
<td>512</td>
<td>9.96</td>
<td>0.240</td>
<td>0.423</td>
<td>0.220</td>
<td>9.425</td>
<td>0.248</td>
<td>27.5</td>
<td>9.0</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>16:19:26.016</td>
<td>19</td>
<td>116589.00</td>
<td>56.93</td>
<td>512</td>
<td>10.01</td>
<td>0.236</td>
<td>0.486</td>
<td>0.208</td>
<td>31.922</td>
<td>0.648</td>
<td>27.5</td>
<td>8.1</td>
<td>6.7</td>
<td></td>
</tr>
<tr>
<td>16:19:27.017</td>
<td>20</td>
<td>109109.00</td>
<td>53.28</td>
<td>512</td>
<td>9.98</td>
<td>0.252</td>
<td>0.429</td>
<td>0.233</td>
<td>13.926</td>
<td>0.265</td>
<td>27.5</td>
<td>11.8</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>16:19:28.016</td>
<td>21</td>
<td>123767.00</td>
<td>60.43</td>
<td>512</td>
<td>9.99</td>
<td>0.222</td>
<td>0.448</td>
<td>0.197</td>
<td>15.685</td>
<td>0.566</td>
<td>27.5</td>
<td>8.4</td>
<td>8.5</td>
<td></td>
</tr>
<tr>
<td>16:19:29.030</td>
<td>22</td>
<td>124040.00</td>
<td>60.57</td>
<td>512</td>
<td>9.90</td>
<td>0.225</td>
<td>0.479</td>
<td>0.197</td>
<td>112.039</td>
<td>0.749</td>
<td>27.9</td>
<td>13.2</td>
<td>8.7</td>
<td></td>
</tr>
<tr>
<td>16:19:30.018</td>
<td>23</td>
<td>113921.00</td>
<td>55.63</td>
<td>512</td>
<td>9.94</td>
<td>0.238</td>
<td>0.414</td>
<td>0.191</td>
<td>12.076</td>
<td>0.754</td>
<td>27.2</td>
<td>8.4</td>
<td>7.9</td>
<td></td>
</tr>
</tbody>
</table>
Write Cliff (Preconditioning is over)

**FlashSystem 900: Random Write preconditioning**

Preconditioning 4K Writes - FlashSystem 900 12x5.2 TiB (46.78 TiB, 90%), 32 LUs
16 host to 8 switched 16Gb paths, Two POWER8 S824, AIX 7.1.3.0, PAWs 2.6

Max write IOPS = 700+ K IOPS (sustained over 45 mins)
Long-term write IOPS is about 600 K IOPS

Note that this long-term IO rate is consistent with the 100% 4KB write response time curve in later pages
Phase 1: Data Conditioning and Aging

☸ Conditioning is 2 passes:
  - 5:1 reducible (compressible)
  - Sequential small writes randomized in size around average expected I/O size
  - 5% of data volume

☸ Aging is 5 passes:
  - 5:1 reducible (compressible)
  - Random small writes randomized in size around average expected I/O size
  - 1% of data volume
Data Conditioning Script

compratio=5
*SD:
sd=default, openflags=o_direct
sd=sd1,lun=\\\PhysicalDrive1
sd=sd2,lun=\\\PhysicalDrive2
sd=sd3,lun=\\\PhysicalDrive3
...
sd=sd37,lun=\\\PhysicalDrive37

*WD: Workload Definitions
wd=wd1,sd=sd*,seekpct=sequential

*RD: Run Definitions
*Will run 1 test IO size with random (4k, 8k, 16k) for approximately 3300 GB
rd=rd1,wd=wd1,iorate=max,elapsed=1950,interval=5,rdpct=0,xfersize=(4k,128k,4k),threads=(16)

Run this 2 runs, the sequential IO specification localizes the data placement across 5% of the system, the 3200 elapsed is based on the IOPS for 16K size divided into the 3.3 GB (5%).
Data Aging Script

compratio=5

*SD:
sd=default, openflags=o_direct
sd=sd1,lun=\..\PhysicalDrive1
sd=sd2,lun=\..\PhysicalDrive2
sd=sd3,lun=\..\PhysicalDrive3
...
sd=sd37,lun=\..\PhysicalDrive37

*WD: Workload Definitions
wd=wd1, sd=sd*, seekpct=random

*RD: Run Definitions
*Will run 1 test IO size with random (4k, 128k, 16k) for approximately 660 GB
rd=rd1, wd=wd1, iorate=max, elapsed=390, interval=1, rdpct=0, xfersize=(4k,128k,16k), threads=(16)

Run this 5 runs, the sequential IO specification localizes the data placement across 5% of the system, the 382 elapsed is based on the IOPS for 16K size divided into the 660 GB (1%).
Phase 2: Baseline Workload Tests

- Once the AFA is brought to a state that is considered a start point, actual testing can start
- The next test is the baseline workload
  - 20/80 read/write ratio, all sequential, mixed block sizes matching your actual distribution, data that exhibits a 5:1 data reduction ratio, and displays some I/O banding with drift
  - 65/35 read/write ratio, all random, mixed block sizes matching your actual distribution, data that exhibits a 5:1 data reduction ratio, and displays some I/O banding with drift
  - 35/65 read/write ratio, all random, mixed block sizes matching your actual distribution, data that exhibits a 5:1 data reduction ratio, and displays some I/O banding with drift
  - 80/20 read/write ratio, all sequential, mixed block sizes matching your actual distribution, data that exhibits a 5:1 data reduction ratio, and displays some I/O banding with drift
Banding and Drift

- Also known as spatial and temporal data
- Spatial is sequential data
- Temporal is data that is accessed near the same time
- Buffering/caching usually deals best with temporal data
 Writes in AFA

- Usually large writes will be broken up
- 4-8K write shards are common
- A large write that would be written sequentially to HDD will be “shot gunned” to all available free blocks across several flash modules due to wear leveling
20/80 and 80/20 Read Percent Sequential Workloads

*SD: Storage Definitions
sd=default, openflags=o_direct
sd=sd1,lun=\\PhysicalDrive1
sd=sd2,lun=\\PhysicalDrive2
sd=sd3,lun=\\PhysicalDrive3
...
sd=sd37,lun=\\PhysicalDrive37

*WD: Workload Definitions
wd=wd1,sd=sd*,seekpct=sequential

*RD: Run Definitions
*Will run 40 tests (10 for 8k, 10 for 16k, 10 for 32k, 10 for 128k) for 120 seconds each
rd.rd1,wd=wd1,iorate=max,elapsed=300,interval=1,pause=5,forrdpct=(20,80),forxfersize=(8k,16k, 32k,128k),forthreads=(1,16,d)
35/65 and 65/35 Read Percent Random Workloads

*SD: Storage Definitions
sd=default, openflags=o_direct
sd=sd1,lun=\\PhysicalDrive1
sd=sd2,lun=\\PhysicalDrive2
sd=sd3,lun=\\PhysicalDrive3
...
sd=sd37,lun=\\PhysicalDrive37

*WD: Workload Definitions
wd=wd1,sd=sd*,seekpct=random

*RD: Run Definitions
*Will run 40 tests (10 for 8k, 10 for 16k, 10 for 32k, 10 for 128k) for 120 seconds each
rd=rd1,wd=wd1,iorate=max,elapsed=300,interval=1,pause=5,forrdpct=(35,65),forxfersize=(8k,16k,32k,128k),forthreads=(1,16,d)
Determine the Inflection Point

Response Time

<table>
<thead>
<tr>
<th>IOPS</th>
<th>Response Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>39753.53</td>
<td>1</td>
</tr>
<tr>
<td>73836.34</td>
<td>2</td>
</tr>
<tr>
<td>102764.82</td>
<td>3</td>
</tr>
<tr>
<td>116606.87</td>
<td>4</td>
</tr>
<tr>
<td>120978.57</td>
<td>5</td>
</tr>
<tr>
<td>115123.09</td>
<td>6</td>
</tr>
</tbody>
</table>

Utilizing VDBench to Perform IDC AFA Testing

Approved SNIA Tutorial © 2015 Storage Networking Industry Association. All Rights Reserved.
A Better Method May be IOPS

35/65 R/W 16kb IOPS

Parallel Threads/LUN

IOPS

0 20000 40000 60000 80000 100000 120000 140000

4 8 16 32 64 128
Phase 3: Functional Testing

- Applies to full SAN type models
- Will be dependent on functions available
  - Thin provisioning
  - Snapshots
  - Replication
  - Encryption – All 840s do this transparently
  - Copy
  - Backup
  - Etc.
- Not covered in this presentation since it is array specific
Phase 4: Fault Injection Testing

- Determine failure scenarios to test
- Establish base line, steady state mixed workload
- Test Scenarios
  - Example Scenarios
    - Fail a path
    - Fail a drive (or flash module)
    - If the array claims to support it, fail two drives (or flash modules) in the same RAID group simultaneously
    - Fail a controller
    - Fail a power supply (and fail a fan separately if the array supports that)
    - Perform a controller firmware upgrade
    - Expand capacity by adding a shelf online
4KB 20/80 load

4kb_20_80_seq.cfg

*SD: Storage Definitions
sd=default, openflags=o_direct
sd=sd1,lun=\\.\PhysicalDrive1
sd=sd2,lun=\\.\PhysicalDrive2
sd=sd3,lun=\\.\PhysicalDrive3
sd=sd4,lun=\\.\PhysicalDrive4
sd=sd5,lun=\\.\PhysicalDrive5
sd=sd6,lun=\\.\PhysicalDrive6
sd=sd7,lun=\\.\PhysicalDrive7
sd=sd8,lun=\\.\PhysicalDrive8
sd=sd9,lun=\\.\PhysicalDrive9
sd=sd10,lun=\\.\PhysicalDrive10

*WD: Workload Definitions
wd=wd1,sd=sd*,seekpct=sequential

*RD: Run Definitions
*Will run 1 test 4kb sequential 2 hours long
rd=rd1,wd=wd1,iorate=max,elapsed=7200,interval=1,forrdpct=(20),forxfersize=(4k),forthreads=(32)
8KB 65/35 Load

8kb_65_35_ran.cfg

*SD: Storage Definitions
sd=default, openflags=o_direct
sd=sd11,lun=\\PhysicalDrive11
sd=sd12,lun=\\PhysicalDrive12
sd=sd13,lun=\\PhysicalDrive13
sd=sd14,lun=\\PhysicalDrive14
sd=sd15,lun=\\PhysicalDrive15
sd=sd16,lun=\\PhysicalDrive16
sd=sd17,lun=\\PhysicalDrive17
sd=sd18,lun=\\PhysicalDrive18
sd=sd19,lun=\\PhysicalDrive19
sd=sd20,lun=\\PhysicalDrive20

*WD: Workload Definitions
wd=wd1,sd=sd*,seekpct=random

*RD: Run Definitions
*Will run 1 test 8kb random 2 hours long
rd=rd1,wd=wd1,iorate=max,elapsed=7200,interval=1,forrdpct=(65),forxfersize=(8k),forthreads=(16)
64 KB 65/35 Load

64kb_65_35_ran.cfg
*SD: Storage Definitions
sd=default, openflags=o_direct
sd=sd21,lun=\PhysicalDrive21
sd=sd22,lun=\PhysicalDrive22
sd=sd23,lun=\PhysicalDrive23
sd=sd24,lun=\PhysicalDrive24
sd=sd25,lun=\PhysicalDrive25
sd=sd26,lun=\PhysicalDrive26
sd=sd27,lun=\PhysicalDrive27
sd=sd28,lun=\PhysicalDrive28
sd=sd29,lun=\PhysicalDrive29
sd=sd30,lun=\PhysicalDrive30

*WD: Workload Definitions
wd=wd1, sd=sd*, seekpct=random

*RD: Run Definitions
*Will run 1 test 64kb 2 hours long
rd=rd1, wd=wd1, iorate=max, elapsed=7200, interval=1, forrdpct=(65), forxfersize=(64k), forthreads=(8)
128 KB 80/20 Load

128kb_80_20_seq.cfg

*SD:    Storage Definitions
sd=default, openflags=o_direct
sd=sd31,lun=\\PhysicalDrive31
sd=sd32,lun=\\PhysicalDrive32
sd=sd33,lun=\\PhysicalDrive33
sd=sd34,lun=\\PhysicalDrive34
sd=sd35,lun=\\PhysicalDrive35
sd=sd36,lun=\\PhysicalDrive36
sd=sd37,lun=\\PhysicalDrive37
sd=sd38,lun=\\PhysicalDrive38
sd=sd39,lun=\\PhysicalDrive39

*WD:    Workload Definitions
wd=wd1,sd=sd*,seekpct=sequential

*RD:    Run Definitions
*Will run 1 test 128kb 2 hours long
rd=rd1,wd=wd1,iorate=max,elapsed=7200,interval=1,forrdpct=(80),forxfersize=(128k),forthrea
ds=(4)
Run Loads

Capture outputs to C:\vdbench\output as follows:

Use 4 different sessions and run one of the following commands in each:

\c:\vdbench\vdbench -f 4kb_20_80_seq.cfg -o
\c:\vdbench\output\day_three_4kb –m 10

\c:\vdbench\vdbench -f 8kb_65_35_ran.cfg -o
\c:\vdbench\output\day_three_8kb –m 10

\c:\vdbench\vdbench -f 64kb_65_35_ran.cfg -o
\c:\vdbench\output\day_three_64kb –m 10

\c:\vdbench\vdbench -f 128kb_80_20_seq.cfg -o
\c:\vdbench\output\day_three_128kb –m 9
Do Failures

For our testing we did the following:

• Fail a path
• Reboot a controller
• Fail a controller (unrack controller)
• Fail a module
• We found the test only required 1.5 hour so we did these additional tests:
  • Fail a PSU (pull out cord)
  • Fail a battery (pull out battery module)
Phase 5: Soak Test

- Use the 4 loads from phase 2
- Run each for 12 hours (48 consecutive hours)
- Review results for performance anomalies
Summary

- Testing and comparing systems can be difficult
- Utilizing IDC guidelines testing is made more consistent
- Using tools like VDBench provides a repeatable and consistent test framework
The SNIA Education Committee thanks the following Individuals for their contributions to this Tutorial.

Authorship History
Michael Ault, IBM:
Updates: Incorporated comments 7/22/2015

Additional Contributors
Joseph White
Thomas Rivera

Please send any questions or comments regarding this SNIA Tutorial to tracktutorials@snia.org