



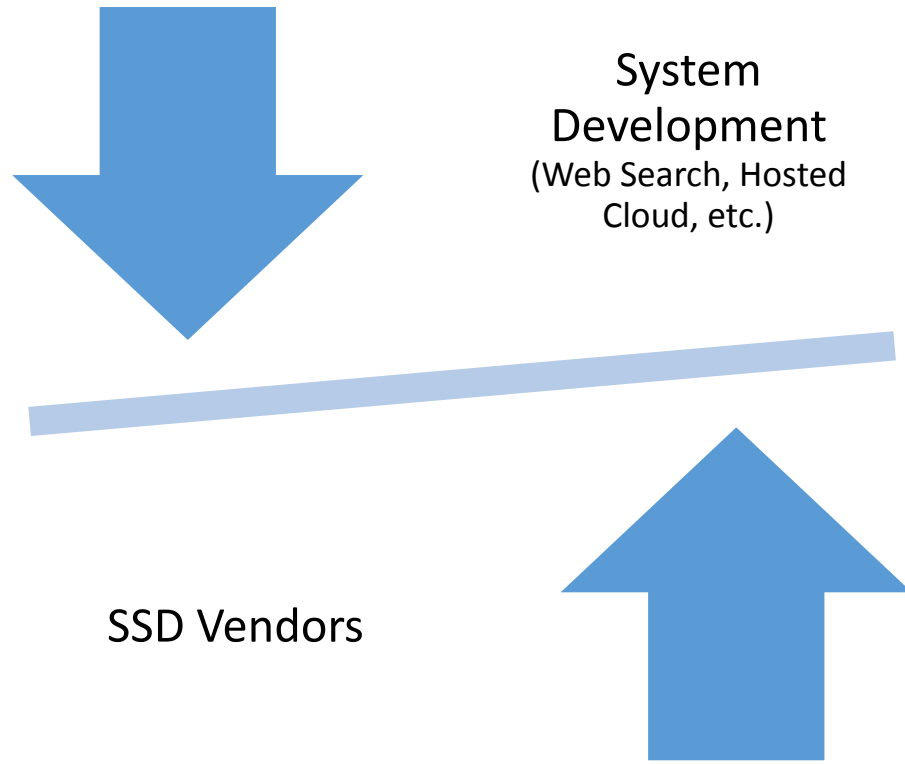
StorScore

Microsoft's System for SSD Qualification

Dr. Laura Caulfield, Mark Santaniello, Dr. Bikash Sharma
Cloud Server Infrastructure Engineering (CSI)



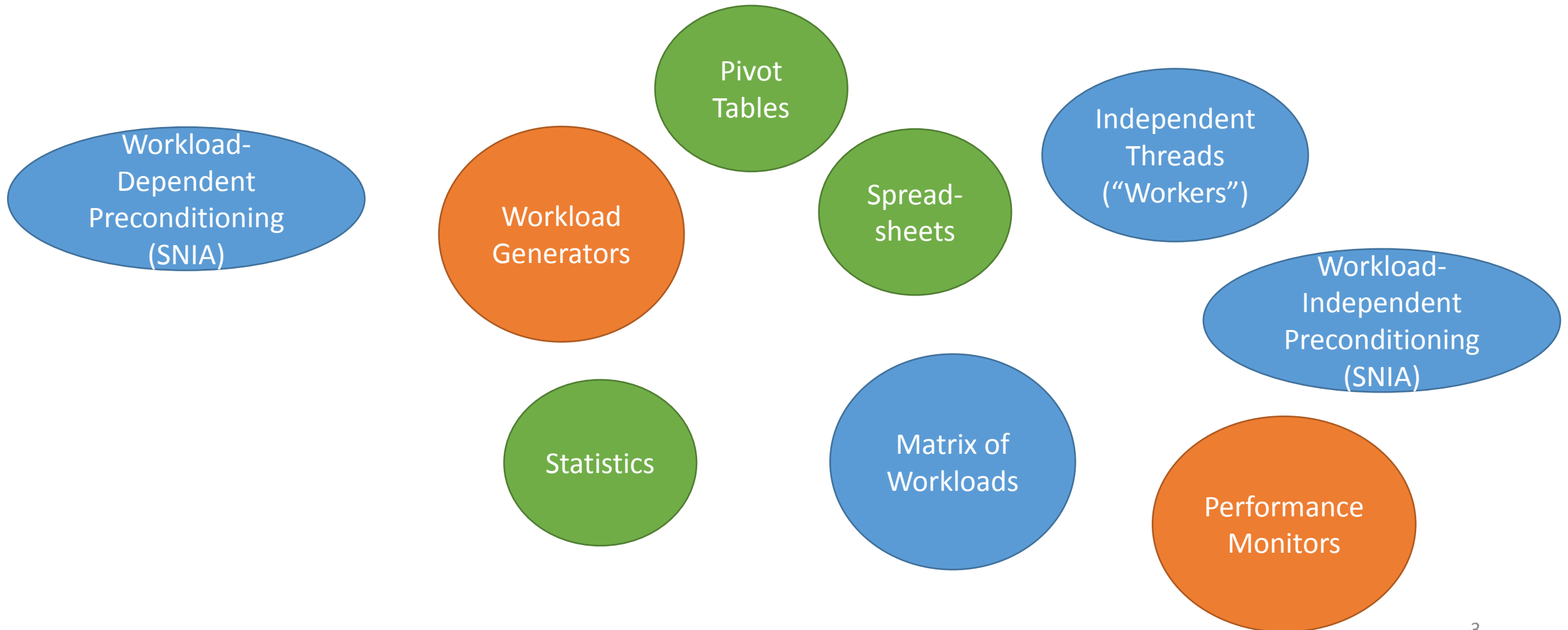
Who are we?



Unique Needs & Opportunities

- Microsoft's platform
- Workloads: Variety and Quantity
- Flexibility to modify stack
- Iterate on designs with vendors
- Wide variety of expertise
- Additional metrics

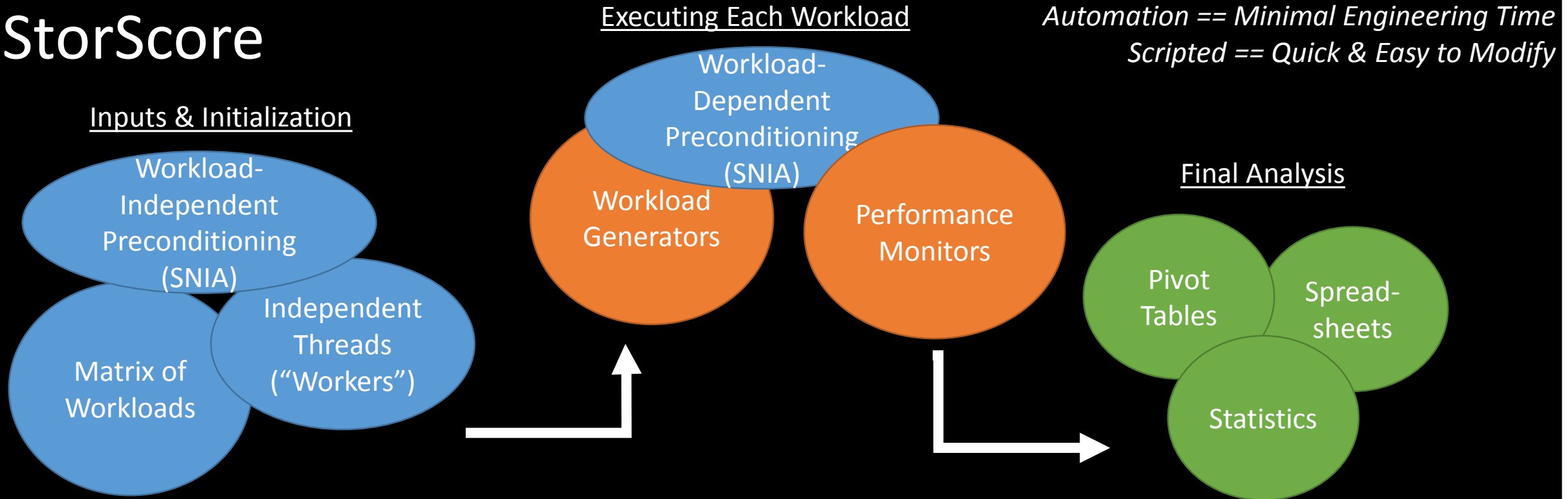
Many Resources & Concepts



What is StorScore?

StorScore is a script wrapper that automates industry-wide best practices for SSD performance testing, existing tools that are under active development for Windows and modern tools and techniques for data analysis.

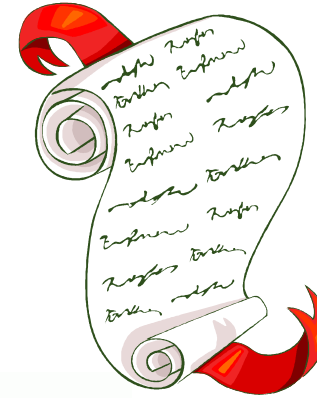
StorScore



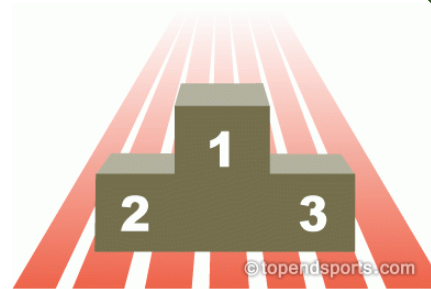


Outline

- ***Recipes***: Defining the Test Suite



- ***Scores***: Managing the Output

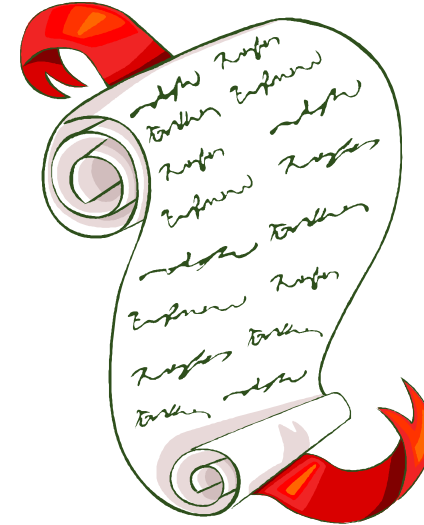


- ***Endurance***: Quantifying the Consumable



Outline

- ***Recipes***: Defining the Test Suite
- ***Scores***: Managing the Output
- ***Endurance***: Quantifying the Consumable





A Single Test

```
test(  
    name_string      => 'foo',  
    write_percentage => 0,  
    access_pattern   => 'random',  
    block_size       => '8K',  
    queue_depth      => 32,  
    warmup_time      => 60,  
    run_time         => 3600  
);
```

- The entire contents of single.rcp



- Reference the file from the cmd line:
\$> StorScore --recipe=single.rcp
- Reads like English



A Matrix of Tests

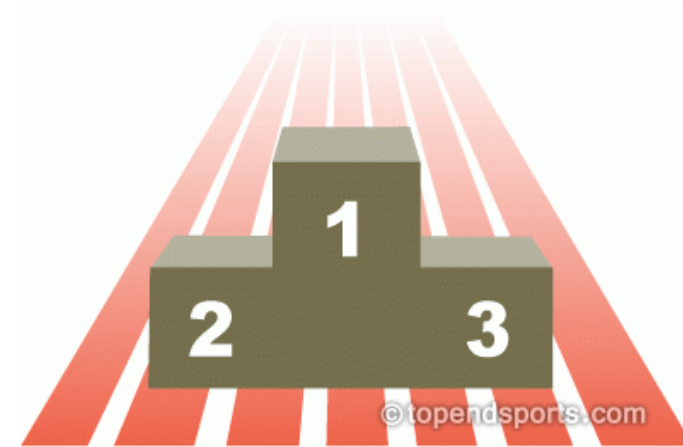
```
# vim: set filetype=perl:  
require 'matrix.rpm';  
  
do_matrix(  
    access_patterns    => [qw( sequential random )],  
    write_percentages => [qw( 100 30 0 )],  
    block_sizes       => [qw( 2M 1M 512K 64K 16K 8K 4K 1K )],  
    queue_depths      => [qw( 256 64 16 4 1 )],  
    warmup_time       => 60,  
    run_time          => 3600  
);  
  
include 'targeted_tests.rcp';
```

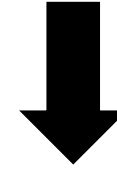
- Mimics Test designer's whiteboard sketch
- “include” statements combine test files
- Full functionality of Perl

```
do_workload( "Targeted Test Read Baseline" );  
  
bg_exec( "smart_loop.cmd $gc{'target_physicaldrive'}" );  
do_workload( "Targeted Test SMART Read Data " );  
bg_killall();
```

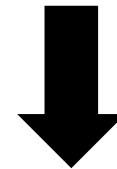

Outline

- *Recipes*: Defining the Test Suite
- *Scores*: Managing the Output
- *Endurance*: Quantifying the Consumable





Display Name	Write Mix	Access Size (kB)	Access Type	Queue Depth	Bandwidth (MB/s)	Average Latency (ms)
Device A	100%	16	random	1	54.32	1.04
Device B	100%	16	random	1	15.05	0.29
Device A	30%	16	random	1	20.01	1.39



Example Policy:

Bandwidth matters a lot, latency matters a little

Device A scores 72/100

Device B scores 65/100

- Raw Output Files → One Excel File
(24 SSDs x 218 Workloads = 5,232 Files)
- Detects and highlights outliers
- Generate Pivot Tables & Graphs
- Still too much data
(5,232 Files x 23 Metrics = 120k Data Pts.)



Putting the “Score” in StorScore

- Goal: Enable data-driven decisions throughout the company
- Reduce data to one score per drive
 - Explainable
 - Repeatable
 - Representative
- Method: a weighted average of all the metrics for each workload

Display Name	Write Mix	Access Size (kB)	Access Type	Queue Depth	Bandwidth (MB/s)	Average Latency (ms)
Device A	100%	16	random	1	Z_AX0	Z_AX1
Device B	100%	16	random	1	Z_BX0	Z_BX1
Device A	30%	16	random	1	Z_AY0	Z_AY1

Step 1:
Convert each value to z-score



Calculating Each Z-Score

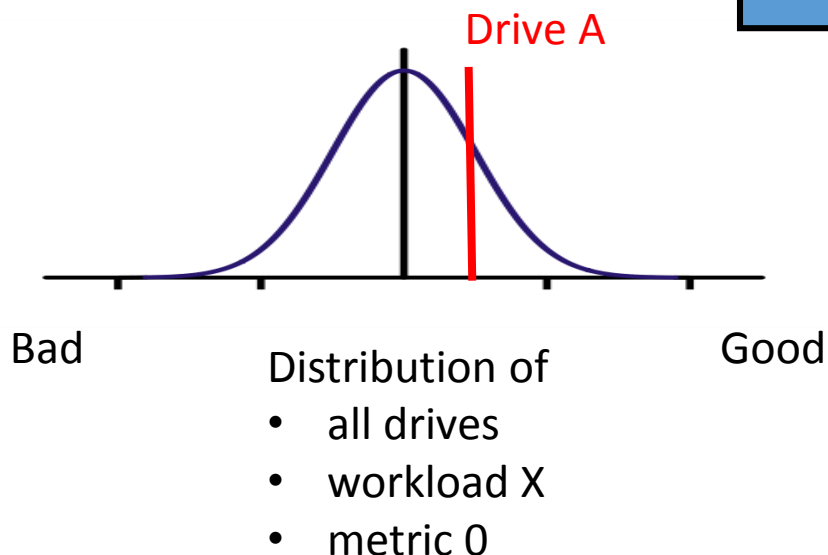
A z-score (or standard score) is the number of standard deviations from the mean.

Drive: A

Wkld: X (4k, rand, QD = 1, 100% writes)

Metric: 0 (Read Latency)

z_{AX0}



- One z-score for each data point
- Positive = better than average
- Negative = worse than average
- Based on cohort of drives



Calculating the Weighted Average

General Policy:

Throughput Metrics

$$50\% \times \boxed{z_{A(n+m)i}}$$

+

Latency Metrics

$$50\% \times \boxed{z_{A(n+m)j}}$$

=

$$70 / 100$$

Drive A
Wkld range 0 to (n+m)
Metric range 0 to i

Score for Drive A

- Can apply multiple policies at once
- Can use any kind of weight system (stay consistent within single policy)

Policy to Favor Mixed Workloads:

70/30 Read/Write Mix Workloads

$$5 \times \boxed{z_{An(i+j)}}$$

+

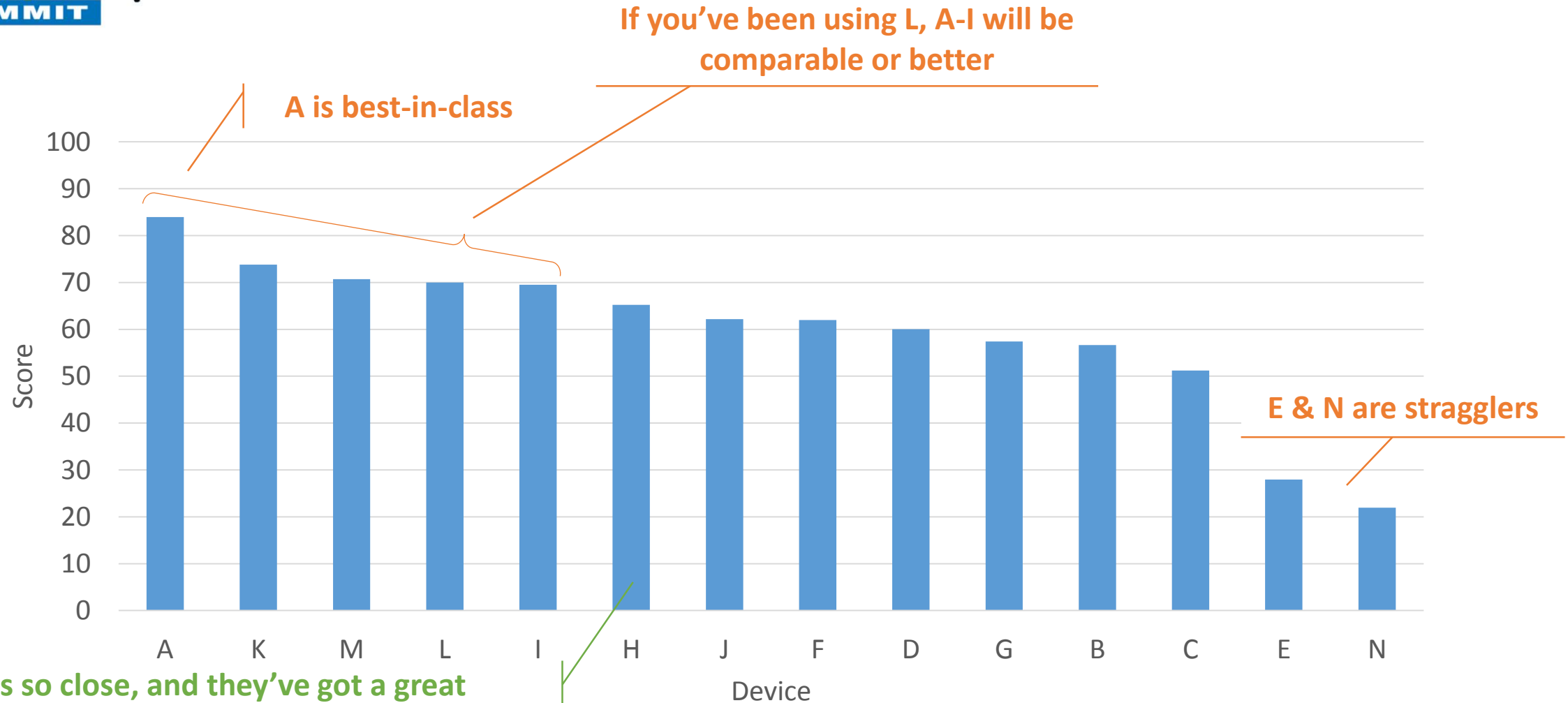
100% Read & 100% Write Workloads

$$1 \times \boxed{z_{Am(i+j)}}$$

=

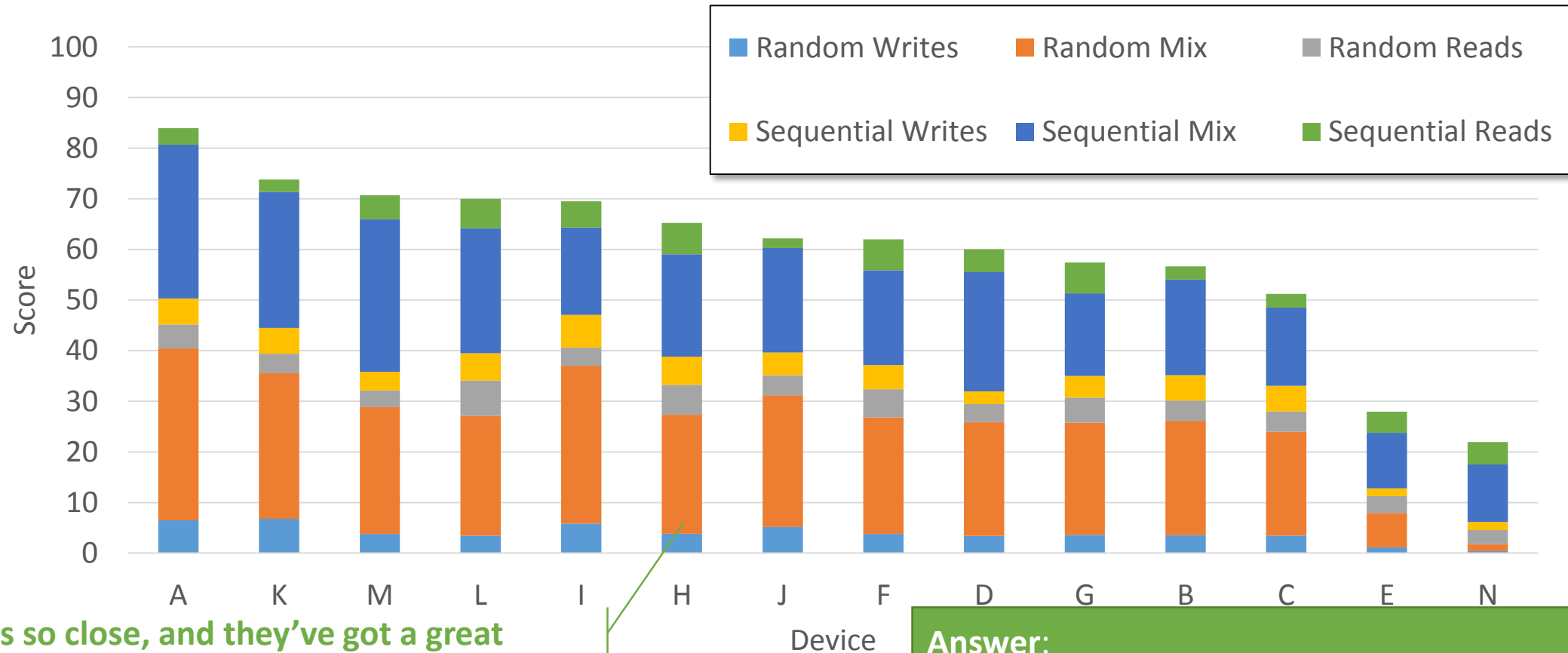
$$65 / 100$$

Scores



H is so close, and they've got a great price. How do we tweak the drive or application to make it work?

Scores' Breakdown



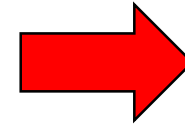
H is so close, and they've got a great price. How do we tweak the drive or application to make it work?

Answer:

Drive should improve random mix (not seq. mix), or App should favor sequential mix (not random mix)

Outline

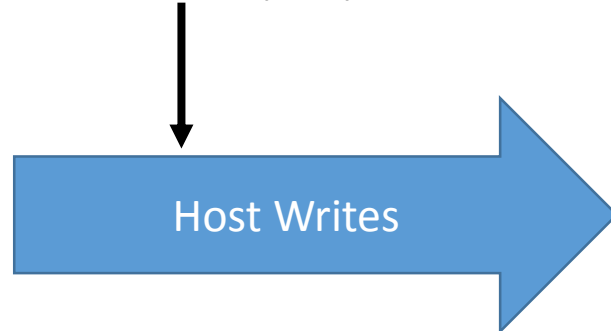
- *Recipes*: Defining the Test Suite
- *Scores*: Managing the Output
- *Endurance*: Quantifying the Consumable



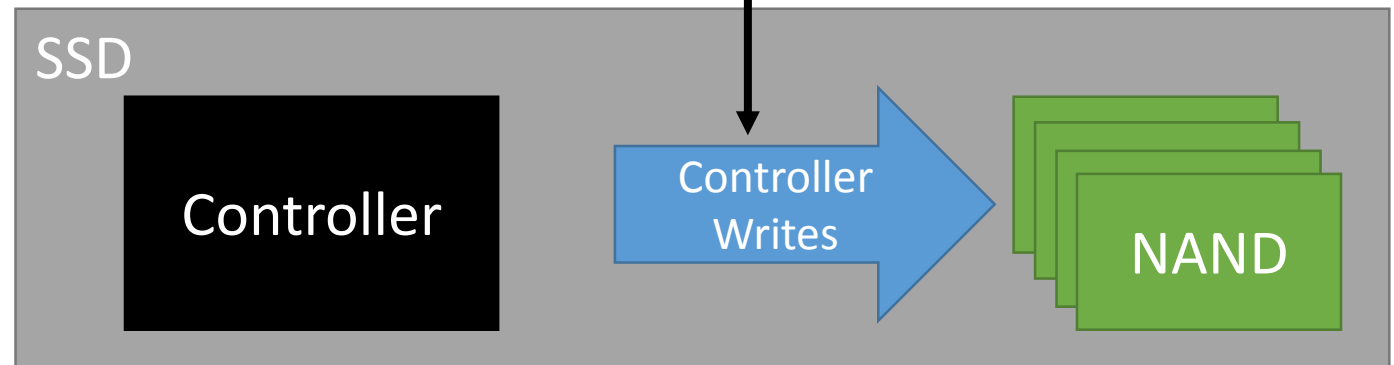


SSD Failure Mechanism: Writes

Drive Writes Per Day (DWPD)
Total Bytes Writes (TBW)
Drive Writes (DW)



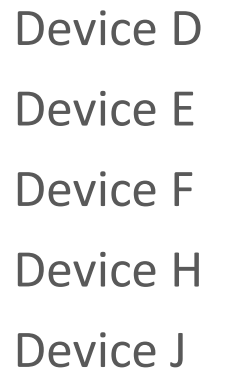
Program / Erase Cycles (P/E Cycles, or PEC)
Write / Erase Cycles (W/E Cycles)



$$\text{Total Drive Writes} \times \text{Write Amplification Factor} = \text{P/E Cycles}$$

Workload Dependent,
Vendor Reported, Implementation Specific

Previously Available	New Telemetry
SMART "Media Wear Indicator"	SMART "Controller Writes"
Reported in units of 1% (300 TB for 30k, 1TB drive)	Reported in units of sectors or GB
4.7 months for 1 workload	1,700 workloads in 4.7 months

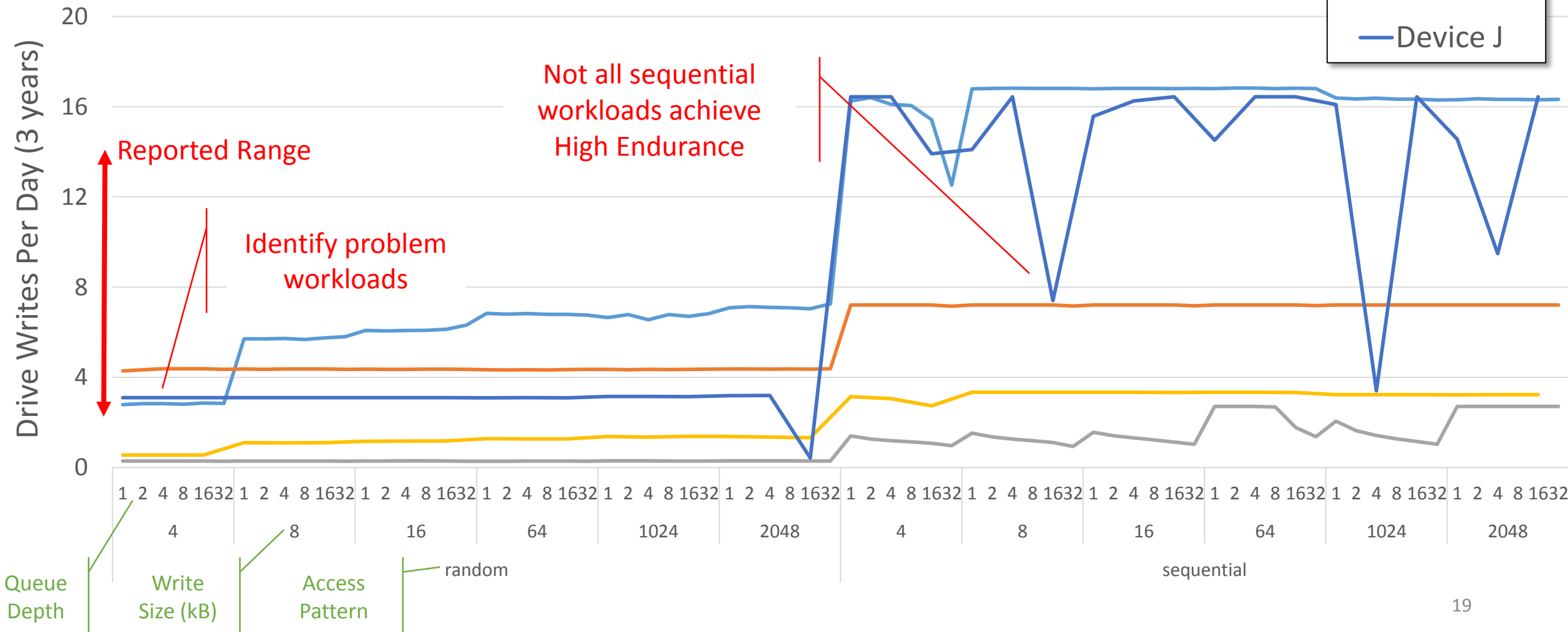




Endurance Results



- Device D
- Device E
- Device F
- Device H
- Device J





Conclusion

- How StorScore brings together existing work & concepts
- Simplicity of defining the inputs
- Spectrum of analysis tools
 - Directly and interactively with excel & pivot charts
 - Automated Score generation
 - Burrowing down into portions of the score
- Measuring endurance on many workloads

StorScore
enables data-driven decision
making process for Microsoft
cloud applications



Thanks! Questions?

You may download StorScore for free at:

<http://aka.ms/storScore>

