Key Attributes and Tools for Testing PCIe Based Solid State Storage

Presenter: Bob Weisickle

CEO/Founder OakGate Technology, Inc

Agenda

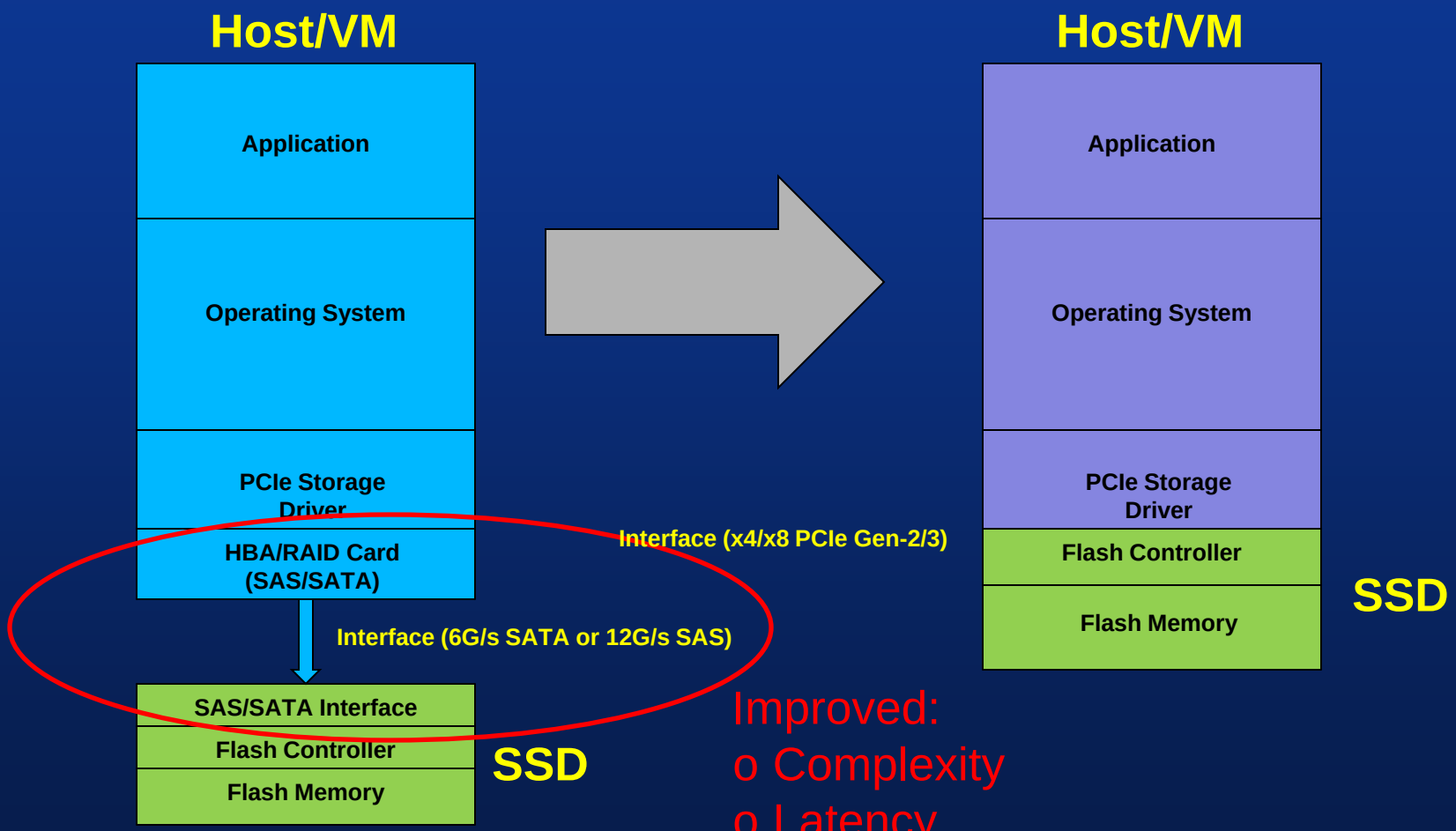
- Premise and Challenge of PCIe Solid State Storage
- Test Environment for Storage Providers and End Users
- What are the unique attributes that distinguishes PCIe SS Storage from SAS/SATA
- Tools and Testing approaches to address some key attributes

PCIe Premise/Challenge

- Better Performance
  - Faster Interface
  - Closer to CPU
- More CONSISTENT performance (esp. Latency)
- Simpler Architecture
- Remove bottlenecks to the Flash

The Premise

PCIe Premise/Challenge



Improved:

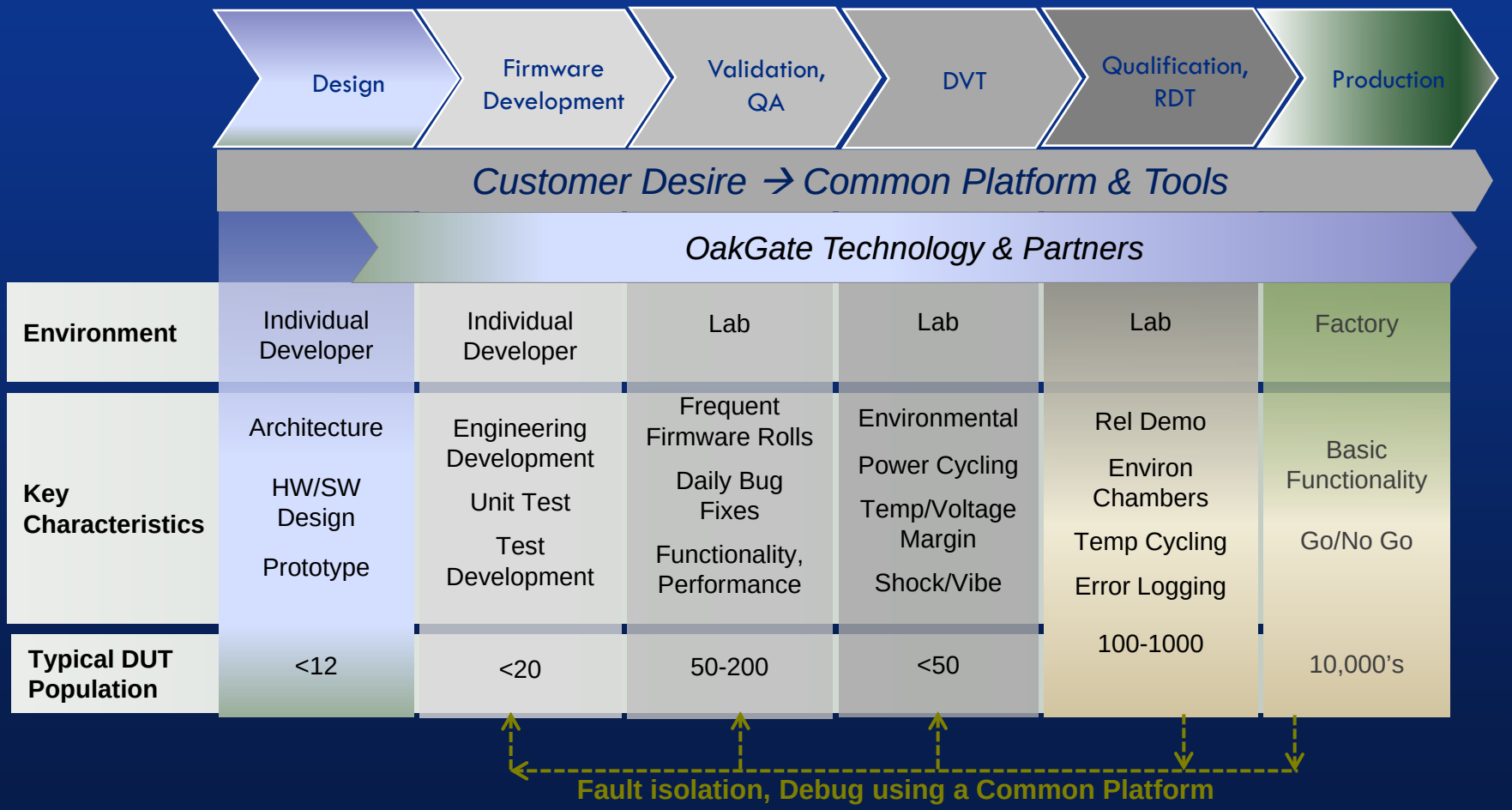
- o Complexity
- o Latency
- o Performance
- o Reliability

PCIe Premise/Challenge

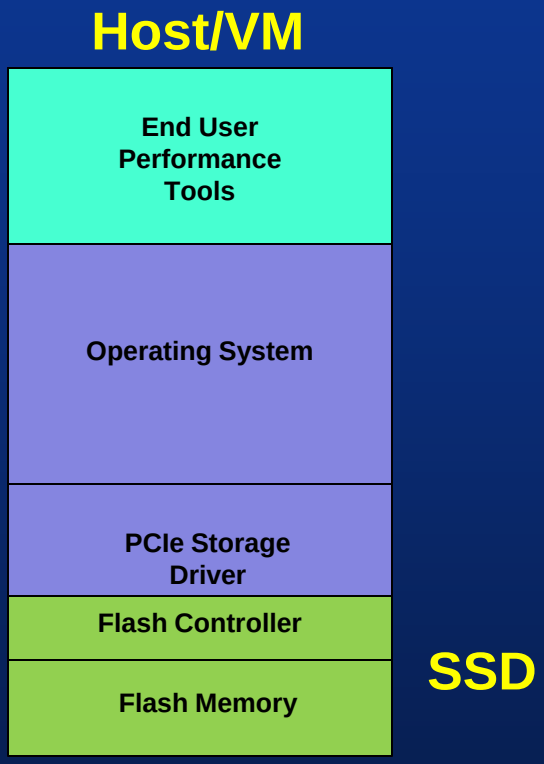
- Multiple/Emerging PCIe Protocols
  - Vendor Proprietary
  - NVMe
  - SCSIe (SCSI over PCIe)
  - SATAe (AHCI)
- New Storage Architecture/Topology
  - PCIe Switching/Bridging
  - New Interfaces/Cables/Enclosures
  - Multi-Host (Physical or Virtual)
  - Multiple Queues
  - Dual Port SFF Architectures
- Higher Performance Levels
  - Tools to generate/measure xMillion IOPS
  - Measure very short IO Latency
  - Scale with Virtual Machines

The Challenge

Environment --- Storage Providers



Environment --- End Users



- Performance Measurement
  - User space Applications like IOmeter
  - Limited ability to tune based on 'real' application workload
- Dependent on Supplier
  - BIOS compatibility
  - IO Failure Analysis
  - Performance Issues
  - Limited means to provide information back to supplier (logs, traces, etc)
  - Suppliers don't 'share' internal tools

Extend the Test Model to encompass the User

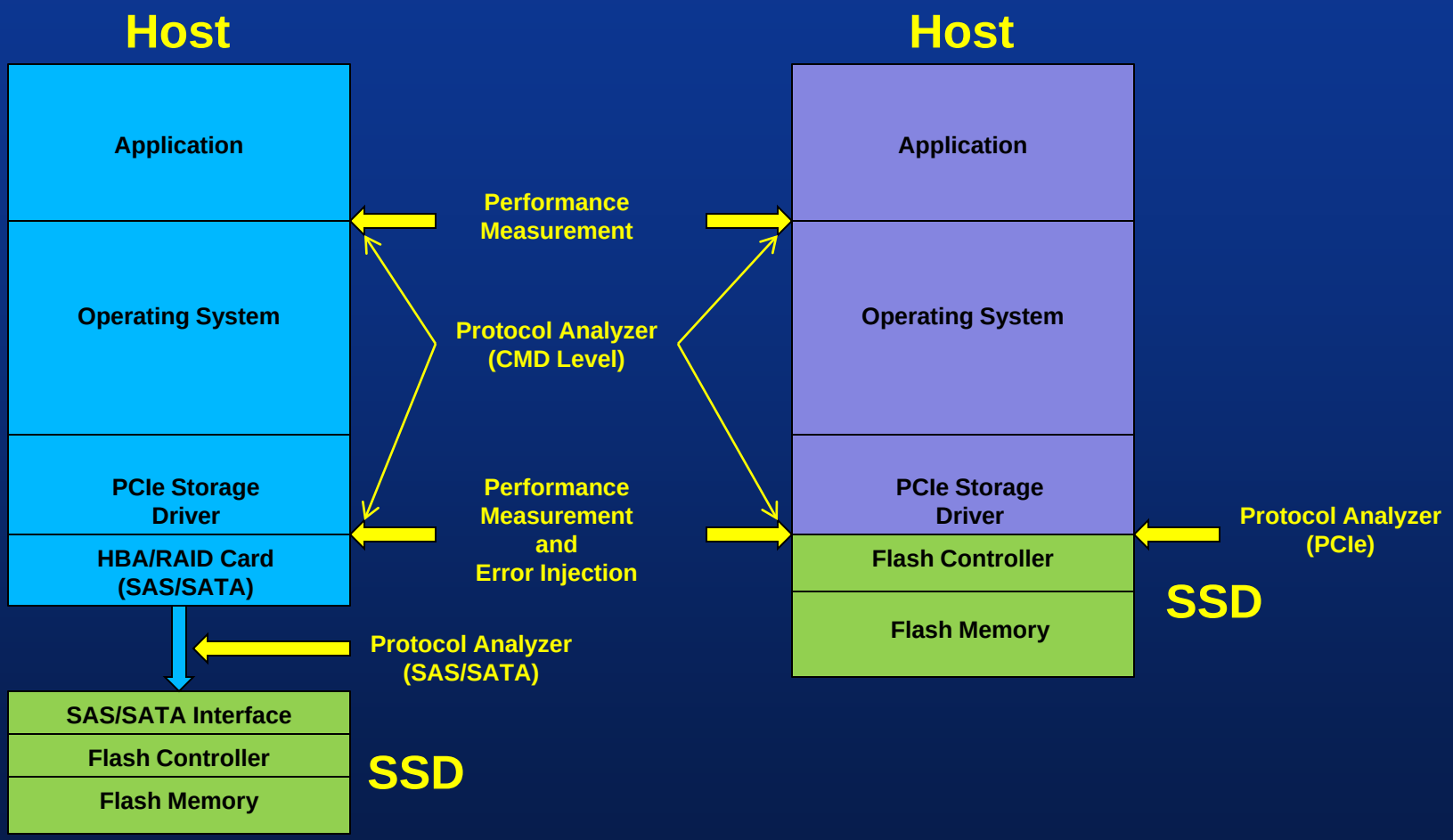
Distinguishing Attributes of PCIe SSD Storage

- What are the unique attributes that distinguishes PCIe SS Storage from SAS/SATA
  - Emerging protocol(s)
  - Maturity (Most are 1st Generation releases)
  - Standards and Proprietary
  - Architecture implementation differences
  - Simplicity of Protocols (opportunity)
- Complexity of Queue'ing model
- Requirements of Tools and Testing approaches
  - Evaluation of HW versus HW + Driver
  - Queue strategy versus Performance
  - Performance Characterization
  - Error Injection Methods

Tools and Testing Approaches

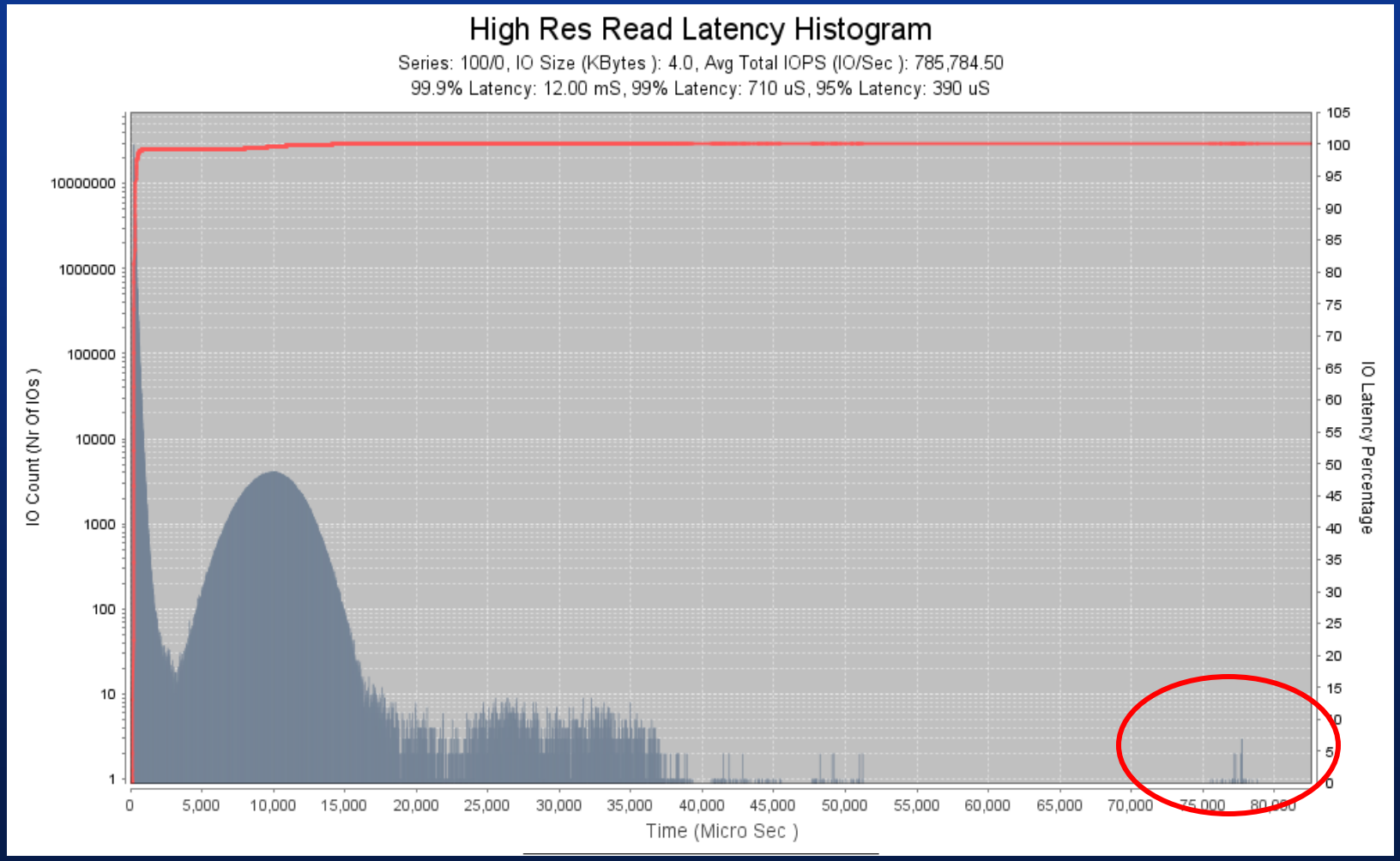
- Performance Measurement
- Impact of Multiple Queues
- Role of Error Injection

Performance Measurement Points



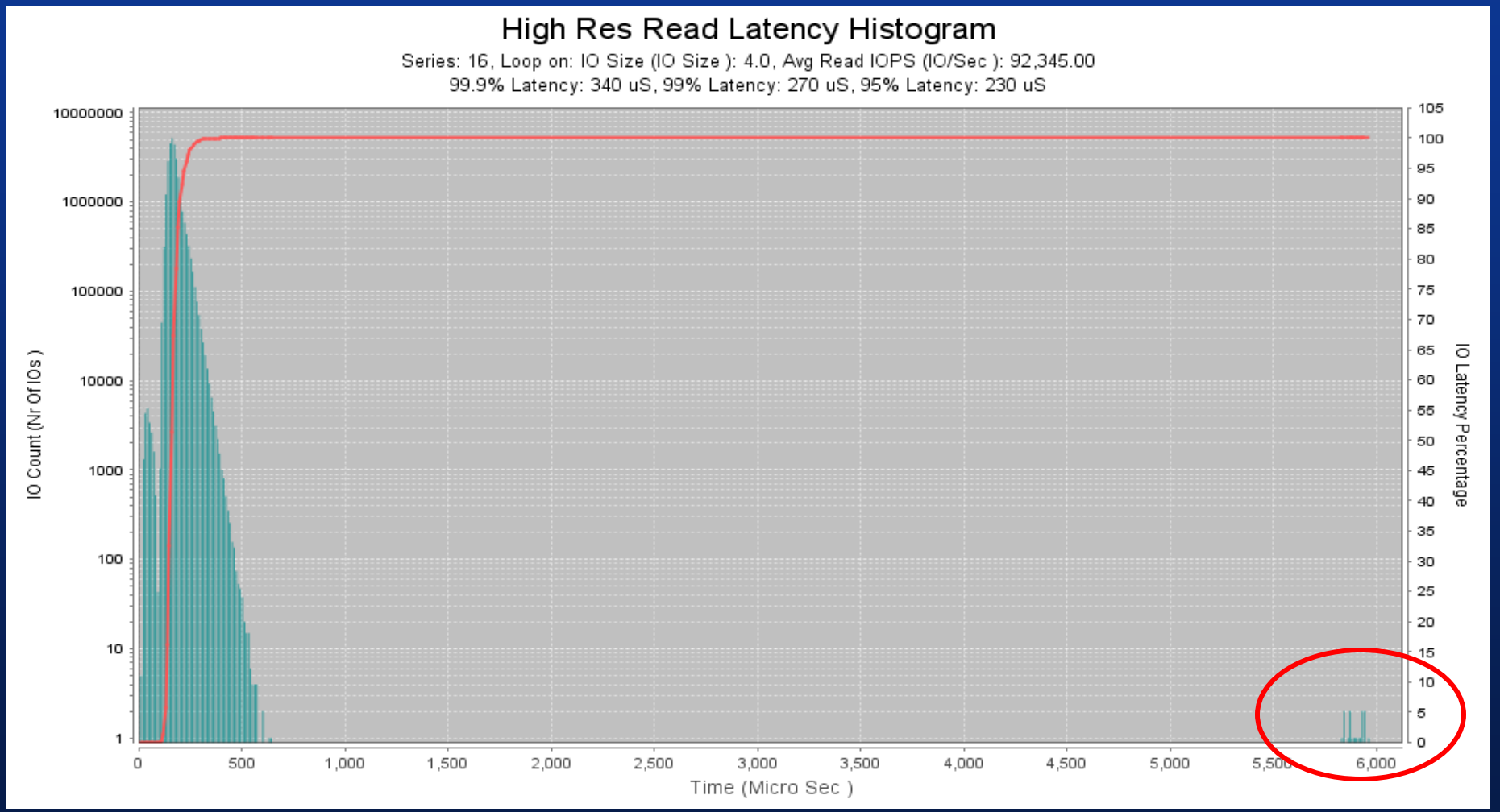
Performance Measurement Examples

Benchmarking



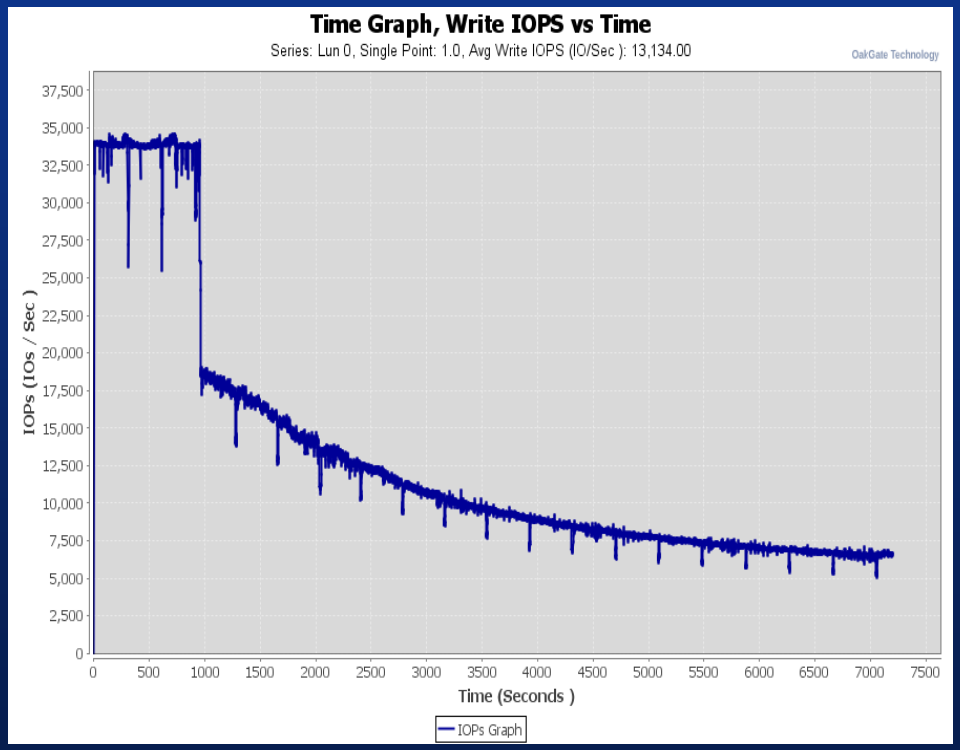
Performance Measurement Examples

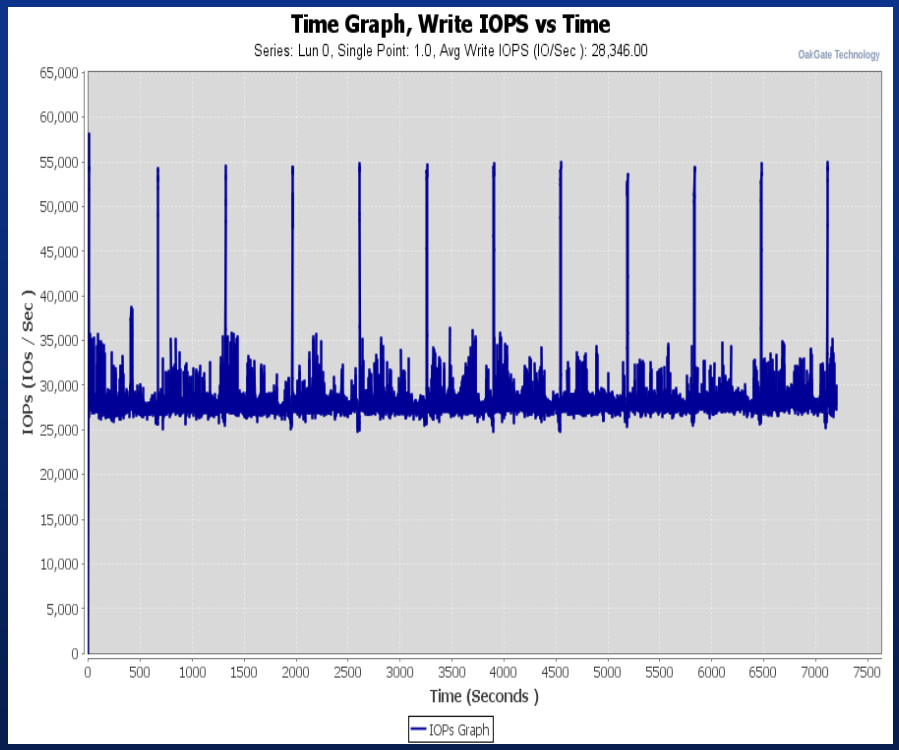
Load Based Performance



Performance Measurement Examples

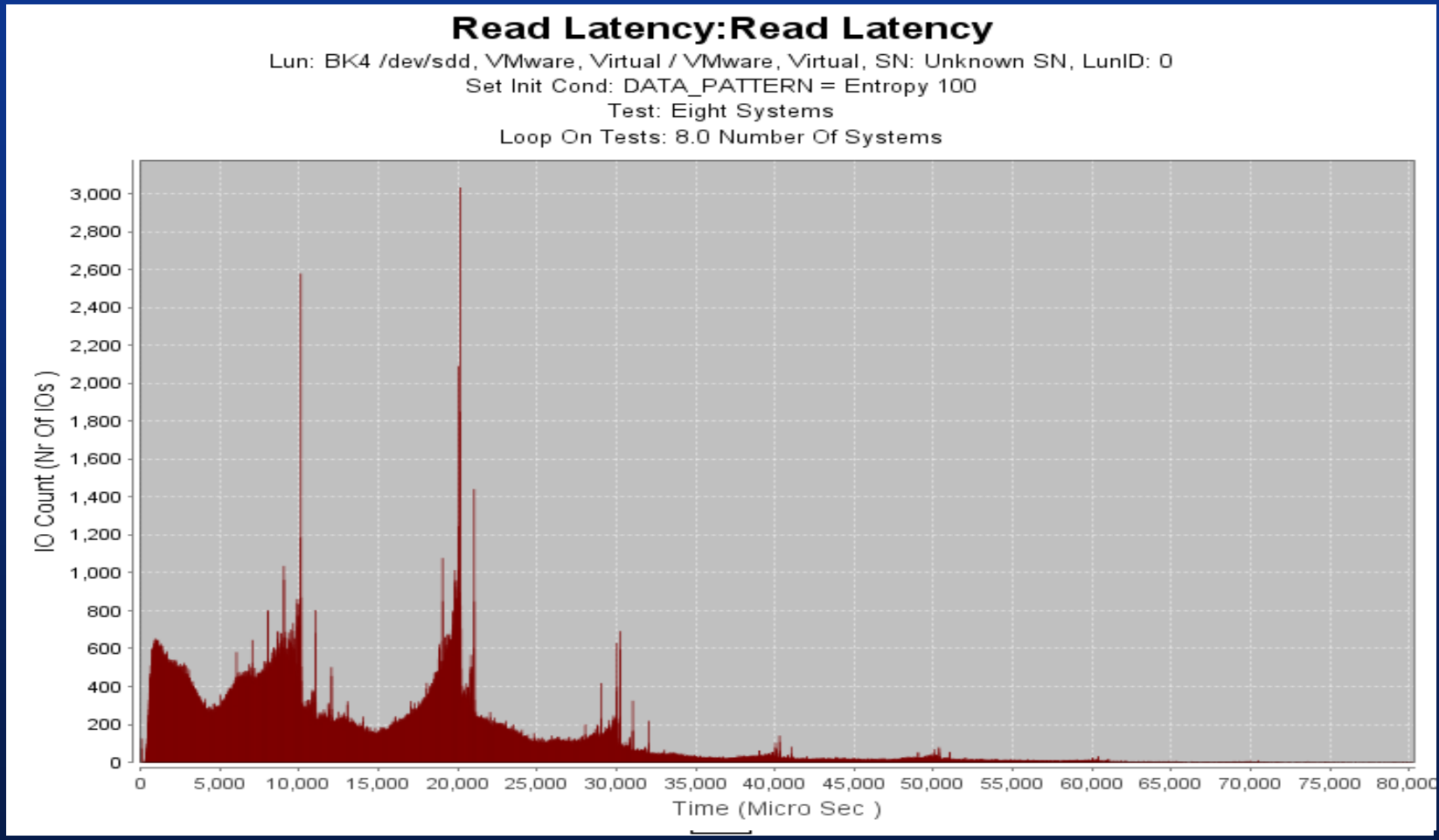
Performance vs Time





Performance Measurement Examples

VM Environment

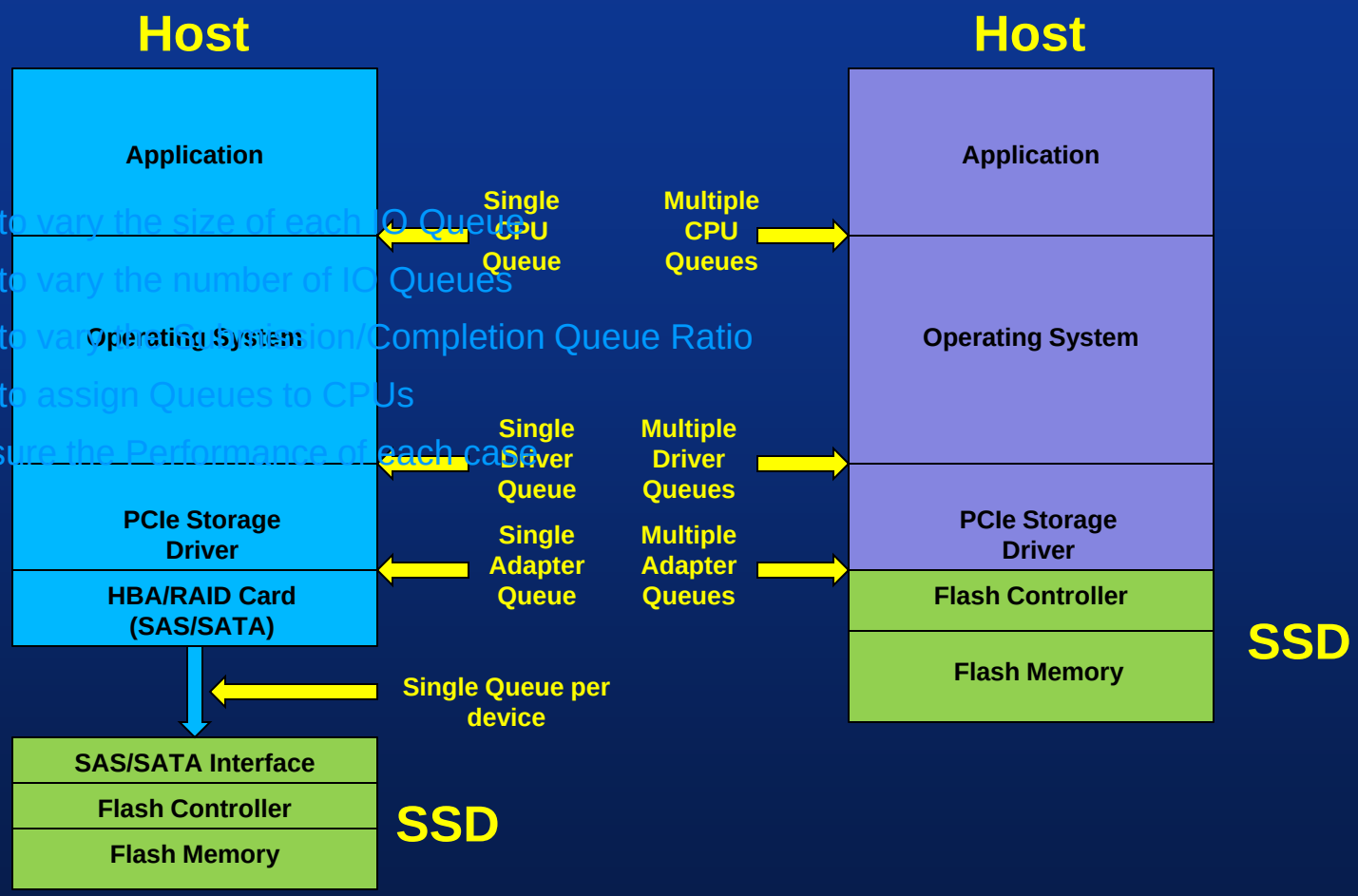


Multiple Queues

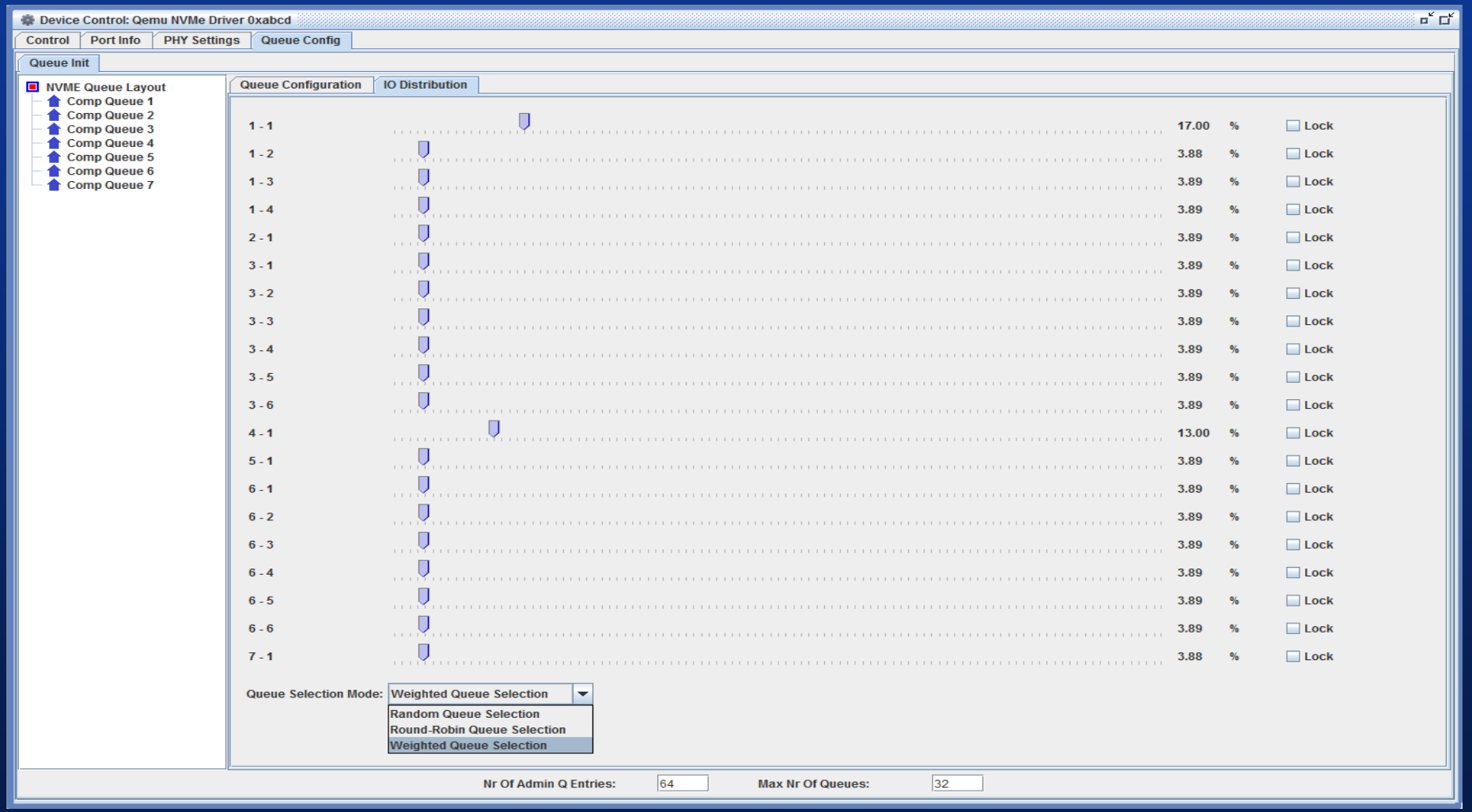
- SAS/SATA used a single Queue to manage IO and Management Commands
- PCIe Proprietary drivers also follow a single Queue model
- Queuing Characteristics of new PCIe Storage Protocol Standards is different
  - Each Storage Device will support multiple Queue sets (IO Submission/Completion)
  - Each Host CPU will have there own Unique Queue sets
  - Management commands have their own unique Queue set
  - Performance may vary with the size and number of Queues used
  - Addressing the needs of Virtual Machines

Multiple Queue Measurement

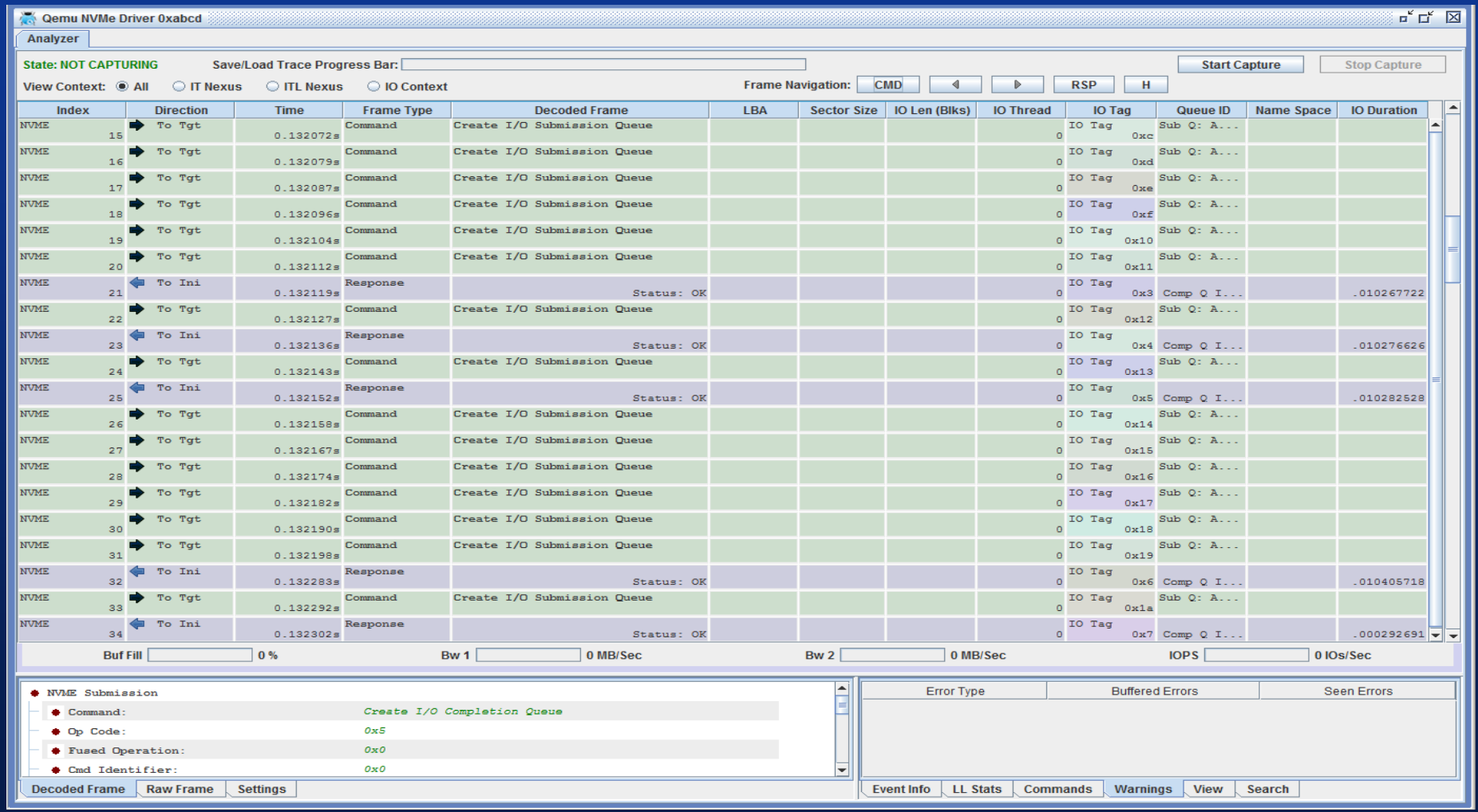
- Able to vary the size of each IO Queue
- Able to vary the number of IO Queues
- Able to vary the Submission/Completion Queue Ratio
- Able to assign Queues to CPUs
- Measure the Performance of each case



NVMe/SCSIe Queue Configuration



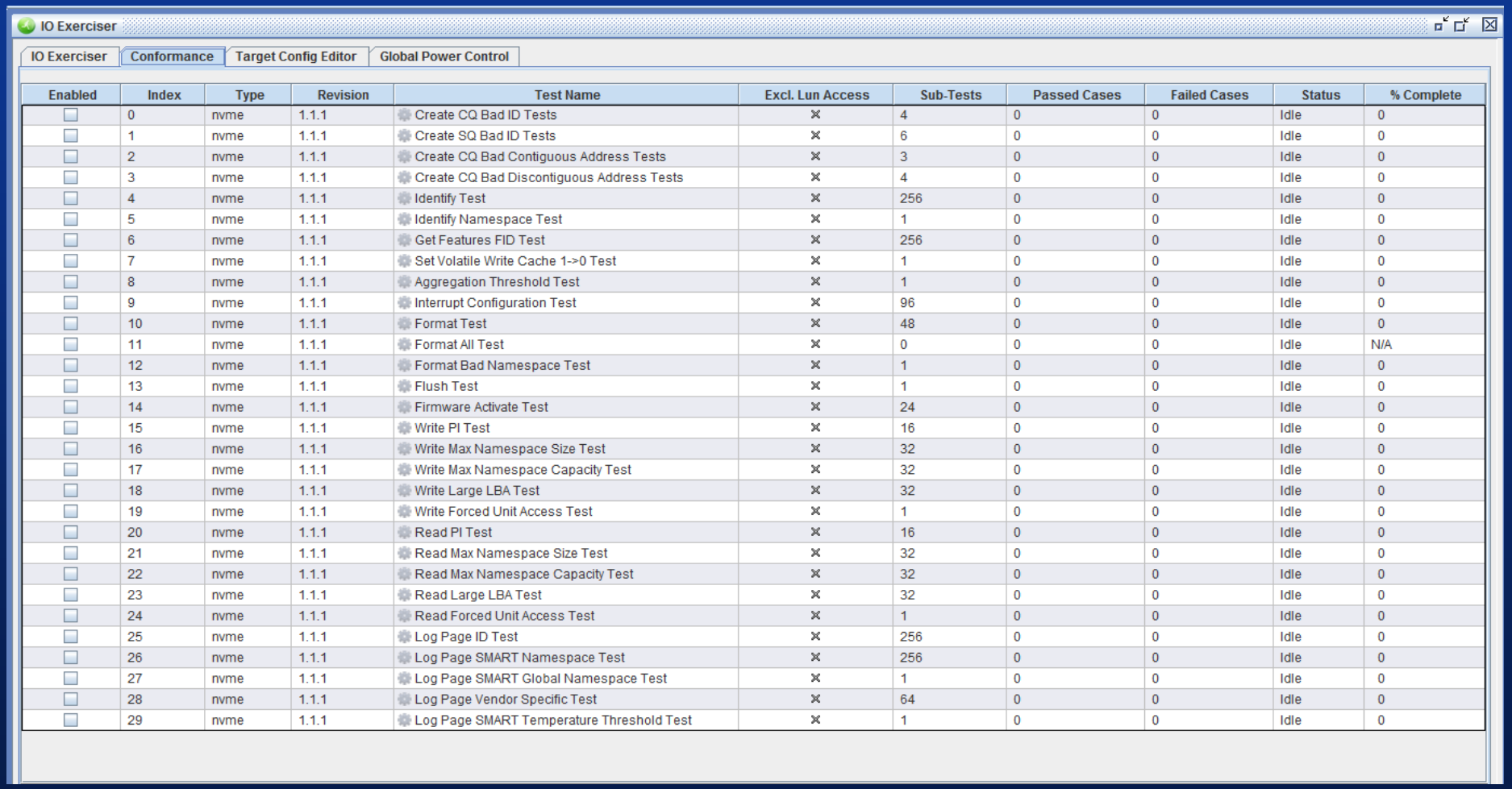
Analyzer Command Verification



Role of Error Injection

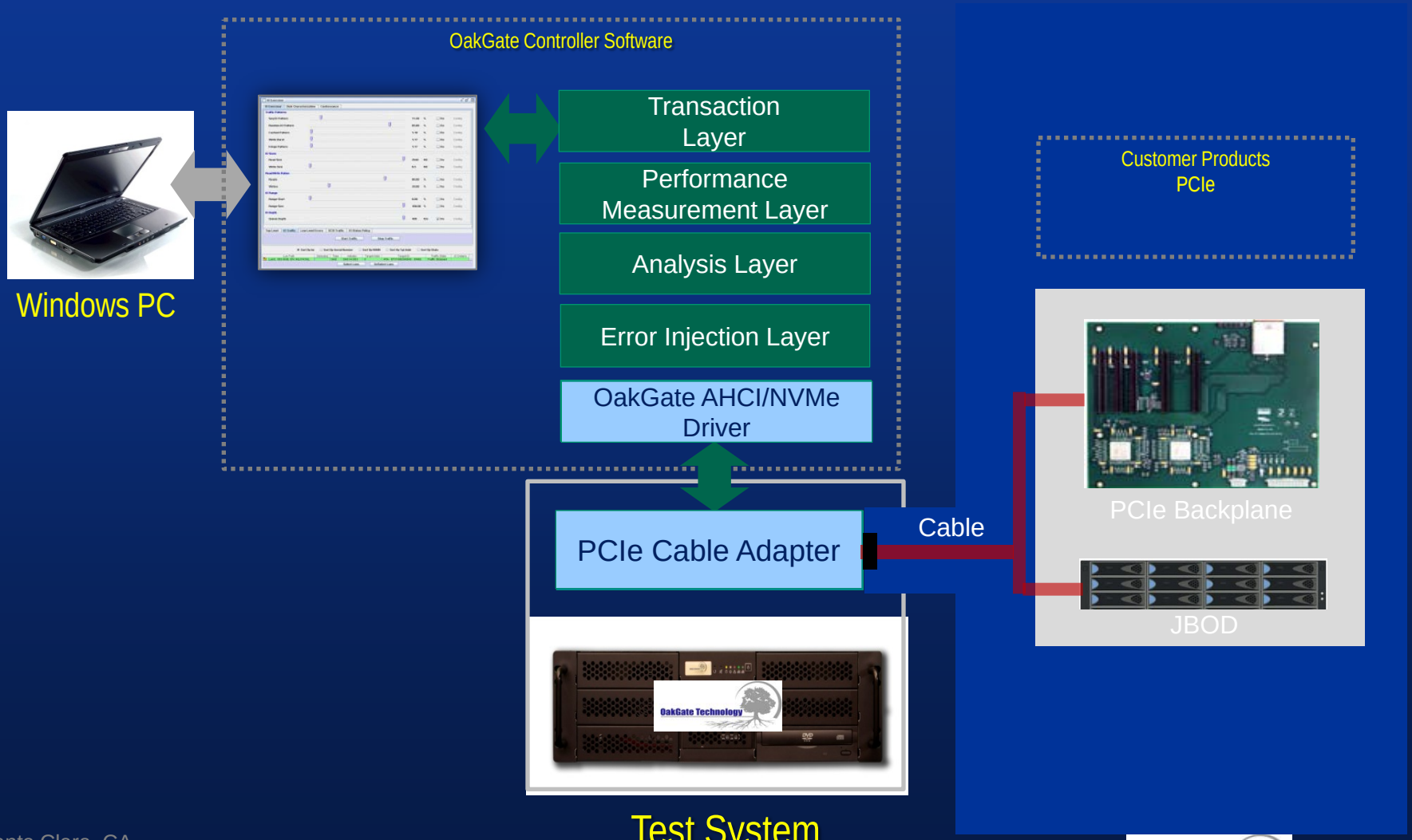
- Focus is to Improve Product Robustness and Stability
  - Increased Code Coverage
  - Verify Error Recovery Paths
  - Exercise full breadth of Product Features
- Mixture of IO traffic with:
  - Management commands
  - Illegal commands
  - Queue creations/deletions
  - Vendor Unique commands
- Data Validation
  - Power Fail testing
  - Write Atomicity
  - Hot Spot Writes
  - Stress Wear Leveling

Directed Error Injection/Validation



Enabled	Index	Type	Revision	Test Name	Excl. Lun Access	Sub-Tests	Passed Cases	Failed Cases	Status	% Complete
☐	0	nvme	1.1.1	Create CQ Bad ID Tests	✕	4	0	0	Idle	0
☐	1	nvme	1.1.1	Create SQ Bad ID Tests	✕	6	0	0	Idle	0
☐	2	nvme	1.1.1	Create CQ Bad Contiguous Address Tests	✕	3	0	0	Idle	0
☐	3	nvme	1.1.1	Create CQ Bad Discontiguous Address Tests	✕	4	0	0	Idle	0
☐	4	nvme	1.1.1	Identify Test	✕	256	0	0	Idle	0
☐	5	nvme	1.1.1	Identify Namespace Test	✕	1	0	0	Idle	0
☐	6	nvme	1.1.1	Get Features FID Test	✕	256	0	0	Idle	0
☐	7	nvme	1.1.1	Set Volatile Write Cache 1->0 Test	✕	1	0	0	Idle	0
☐	8	nvme	1.1.1	Aggregation Threshold Test	✕	1	0	0	Idle	0
☐	9	nvme	1.1.1	Interrupt Configuration Test	✕	96	0	0	Idle	0
☐	10	nvme	1.1.1	Format Test	✕	48	0	0	Idle	0
☐	11	nvme	1.1.1	Format All Test	✕	0	0	0	Idle	N/A
☐	12	nvme	1.1.1	Format Bad Namespace Test	✕	1	0	0	Idle	0
☐	13	nvme	1.1.1	Flush Test	✕	1	0	0	Idle	0
☐	14	nvme	1.1.1	Firmware Activate Test	✕	24	0	0	Idle	0
☐	15	nvme	1.1.1	Write PI Test	✕	16	0	0	Idle	0
☐	16	nvme	1.1.1	Write Max Namespace Size Test	✕	32	0	0	Idle	0
☐	17	nvme	1.1.1	Write Max Namespace Capacity Test	✕	32	0	0	Idle	0
☐	18	nvme	1.1.1	Write Large LBA Test	✕	32	0	0	Idle	0
☐	19	nvme	1.1.1	Write Forced Unit Access Test	✕	1	0	0	Idle	0
☐	20	nvme	1.1.1	Read PI Test	✕	16	0	0	Idle	0
☐	21	nvme	1.1.1	Read Max Namespace Size Test	✕	32	0	0	Idle	0
☐	22	nvme	1.1.1	Read Max Namespace Capacity Test	✕	32	0	0	Idle	0
☐	23	nvme	1.1.1	Read Large LBA Test	✕	32	0	0	Idle	0
☐	24	nvme	1.1.1	Read Forced Unit Access Test	✕	1	0	0	Idle	0
☐	25	nvme	1.1.1	Log Page ID Test	✕	256	0	0	Idle	0
☐	26	nvme	1.1.1	Log Page SMART Namespace Test	✕	256	0	0	Idle	0
☐	27	nvme	1.1.1	Log Page SMART Global Namespace Test	✕	1	0	0	Idle	0
☐	28	nvme	1.1.1	Log Page Vendor Specific Test	✕	64	0	0	Idle	0
☐	29	nvme	1.1.1	Log Page SMART Temperature Threshold Test	✕	1	0	0	Idle	0

OakGate Technology Tool Set



Thank You

Bob Weisickle
bob.weisickle@oakgatetech.com