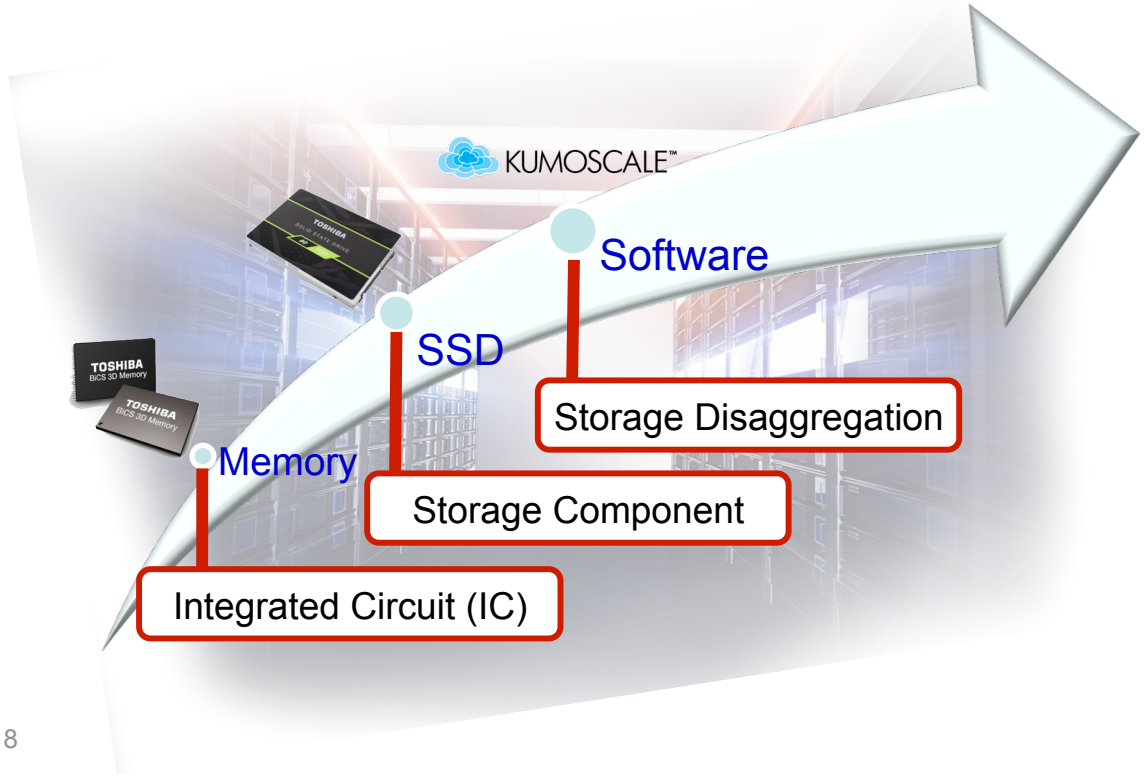
A Kubernetes based Platform for IOT

TOSHIBA

Ram Johri, Systems Architect

About Toshiba Memory Corporation



Challenges with a Scalable Architecture for IOT

Flash Memory Summit

DAS

direct-attached flash

PROBLEM 1

stranded flash & IOPS

PROBLEM 2

stranded compute

PROBLEM 3

lost operational agility & revenue

KumoScale™ enables customers to create an NVMe-oF™ storage node using standard x86 compute platform.

shared accelerated storage

SOLUTION

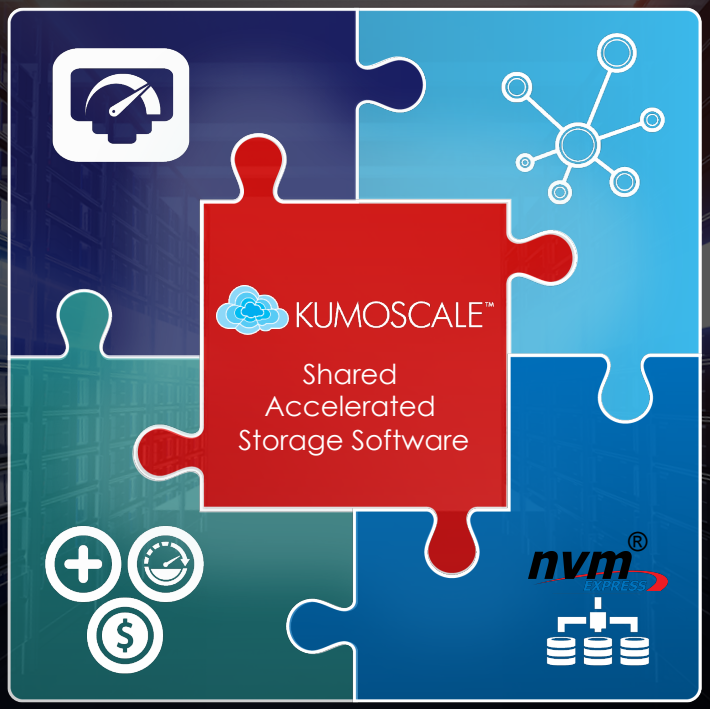
TOSHIBA
Leading Innovation >>>

Shared Accelerated Storage for Cloud Now Possible

Flash Memory Summit

High-bandwidth, low-latency networks

Mature orchestration frameworks



Bigger, faster & more cost-effective NVMe™ SSDs

NVMe-oF™ protocol

Flash Memory Summit 2018
Santa Clara, CA

maximum data center efficiency

TOSHIBA
Leading Innovation

Storage Disaggregation Benchmark Tests

Objective

- Establish NoSQL (MongoDB™) YCSB Benchmark for KumoScale versus DAS versus Competitive NVMe SSDs on
- Bare metal DB hosts
- Containerized Database hosts (docker : openshift)

Goals

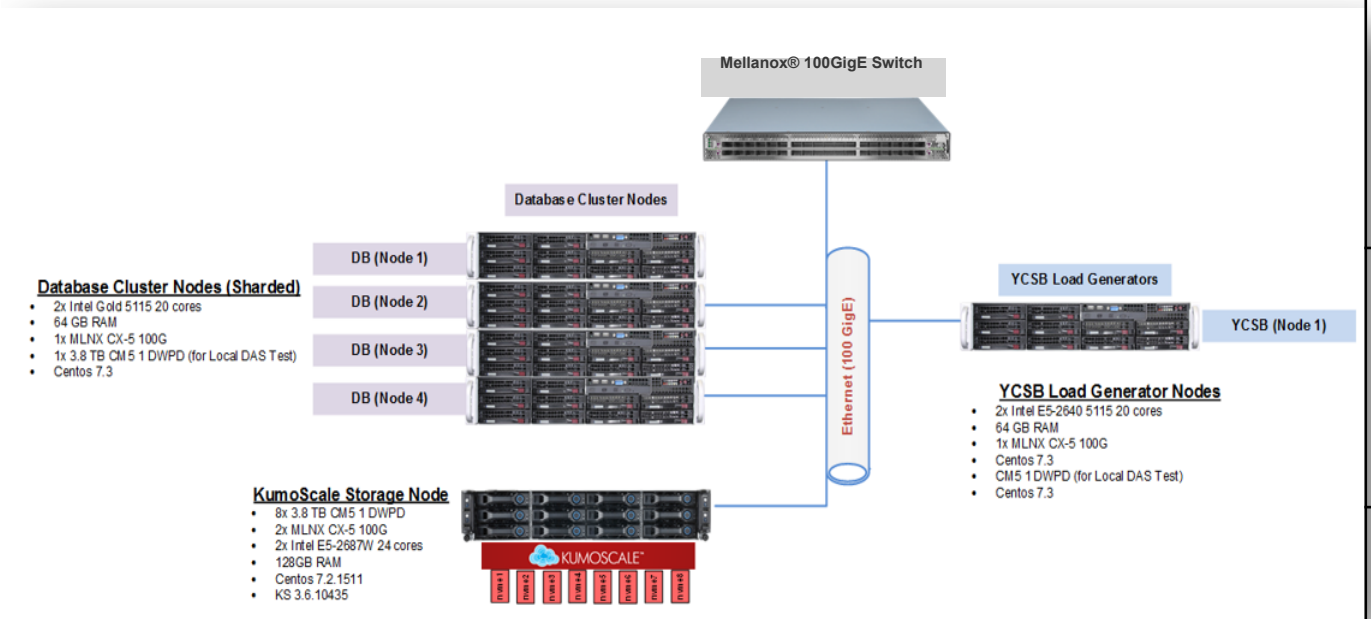
- Bare Metal
- KumoScale is within <10% latency adder over DAS NVMe
- Greater performance

Containerized

- Demonstrate Failover of Compute Node without service disruption
- Demonstrate Failover of Storage Node without service disruption
- Container Storage carved out of 1 KS Node/2 subsystems or 2 KS nodes

Benchmark Test Setup – YCSB: MongoDB™

Flash Memory Summit



Test Profile

YCSB ver: 0.12

MongoDB v3.6.5

300M-500M records of 1KB each

Key-size = 100-bytes

- 3 Shards
- Workload A (Mixed) 50% read + 50% update

Metrics

- Avg. Read Latency
- Avg. Update Latency
- 99th %tile Read Latency
- 99th %tile Update Latency
- Throughput Ops/s

Results

YCSB metrics for:

- Mongo on KumoScale vs
- Mongo on Local NVMe
- Mongo on Local SATA SSD

TOSHIBA
Leading Innovation

Clustered MongoDB™ YCSB Results: Test 1 - 50r:50w Mixed workloadA

**Mongo and YCSB Not Tuned for Performance*

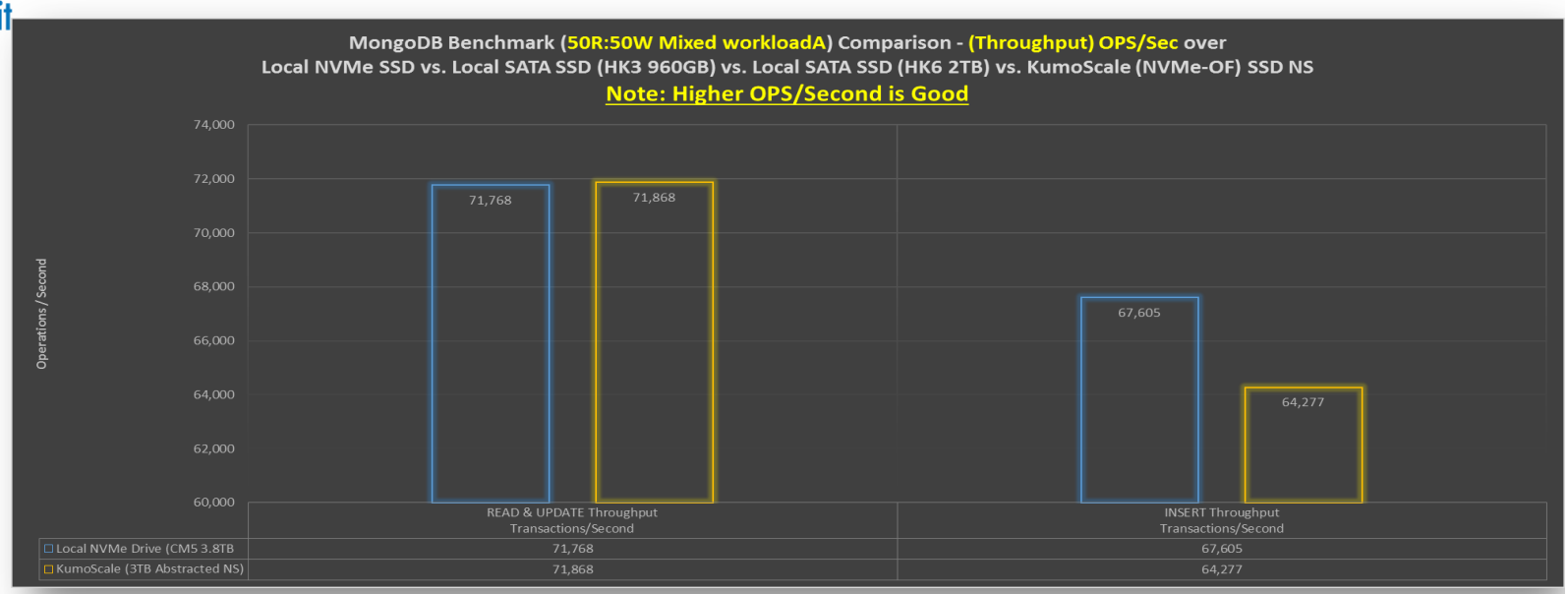
Metrics	Local NVMe (CM5 3.8T)	KumoScale NVMe-oF (3T Abstracted NS)
Configuration	3 MongoDB Shards : 1 MongoS router : 1 YCSB Engine	
Load Phase: Insert	300 Million records : 1K record size : 200 parallel Threads	
• Average Insert Latency (ms) (lower is good)	2.95	3.10
• 99th percentile Insert latency (ms)	5.15	5.61
• Insert Speed (operations/sec) (higher is good)	67,605	64,277
Run Phase: Read & Update		
• Avg Read Latency (ms)	2.74	2.72
• Avg Update Latency (ms)	2.83	2.83
• 99th percentile Read latency (ms)	5.25	5.18
• 99th percentile Update latency (ms)	5.36	5.28
• Read & Update speed (operations/sec) (higher is good)	71,768	71,868

TOSHIBA
Leading Innovation

MongoDB™ Benchmark (50R:50W Mixed workloadA) Comparison – (INSERT, READ, UPDATE) AVG Latency

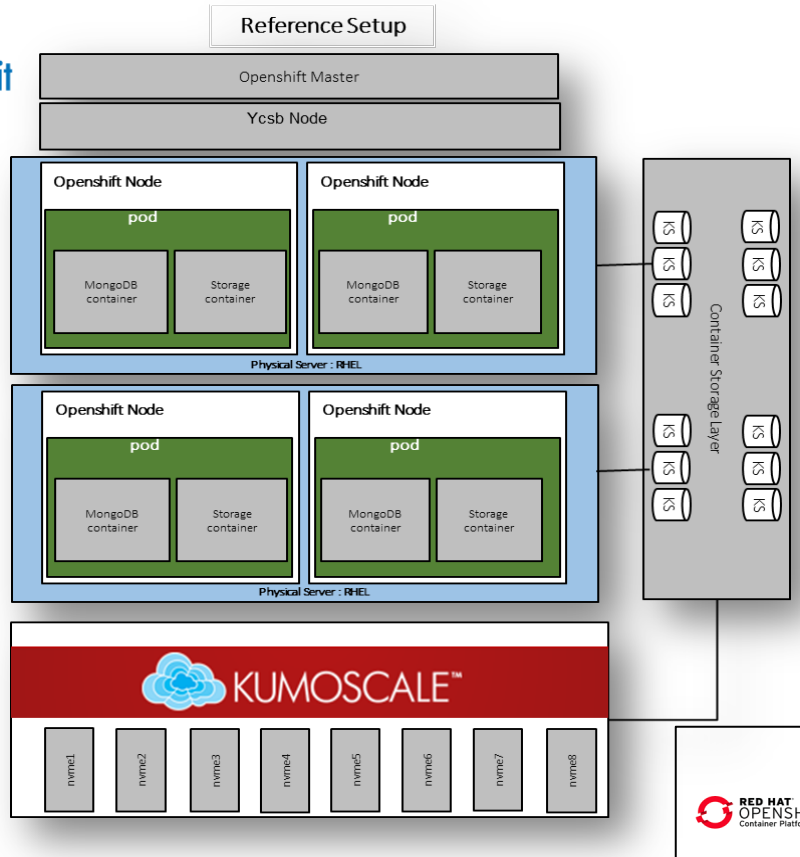
The above chart illustrates the AVG Latency Operations over the given Local Target(NVMe) vs. KumoScale (NVMe-OF) using Mellanox 100GigE Switches. KumoScale Latency shows overall better performance in comparison with others.

MongoDB™ Benchmark (50R:50W Mixed workloadA) Comparison – (INSERT, READ, UPDATE) OPS/Second → Throughput



The above chart illustrates the Operations / Second over the given Local Target(NVMe) vs. KumoScale (NVMe-OF) using Mellanox 100GigE Switches. KumoScale Throughput shows overall better performance in comparison with others.

Test 2: MongoDB™ on RED HAT® OpenShift® Platform



OpenShift Node

- One or more nodes per physical server.
- One or more containers run in one node.

Problem 1: Red Hat Gluster Storage (RHGS)

- Container Storage carved out of 1 KS Node/2 subsystems or 2 KS nodes.
- RHGS dynamically manages pool of storage (KS volumes) to hold data for stateful containers
- RHGS provides replication of data across volumes.
- Demonstrate Failover of Storage Node without service disruption

Problem 2 & Problem 3: OpenShift Master

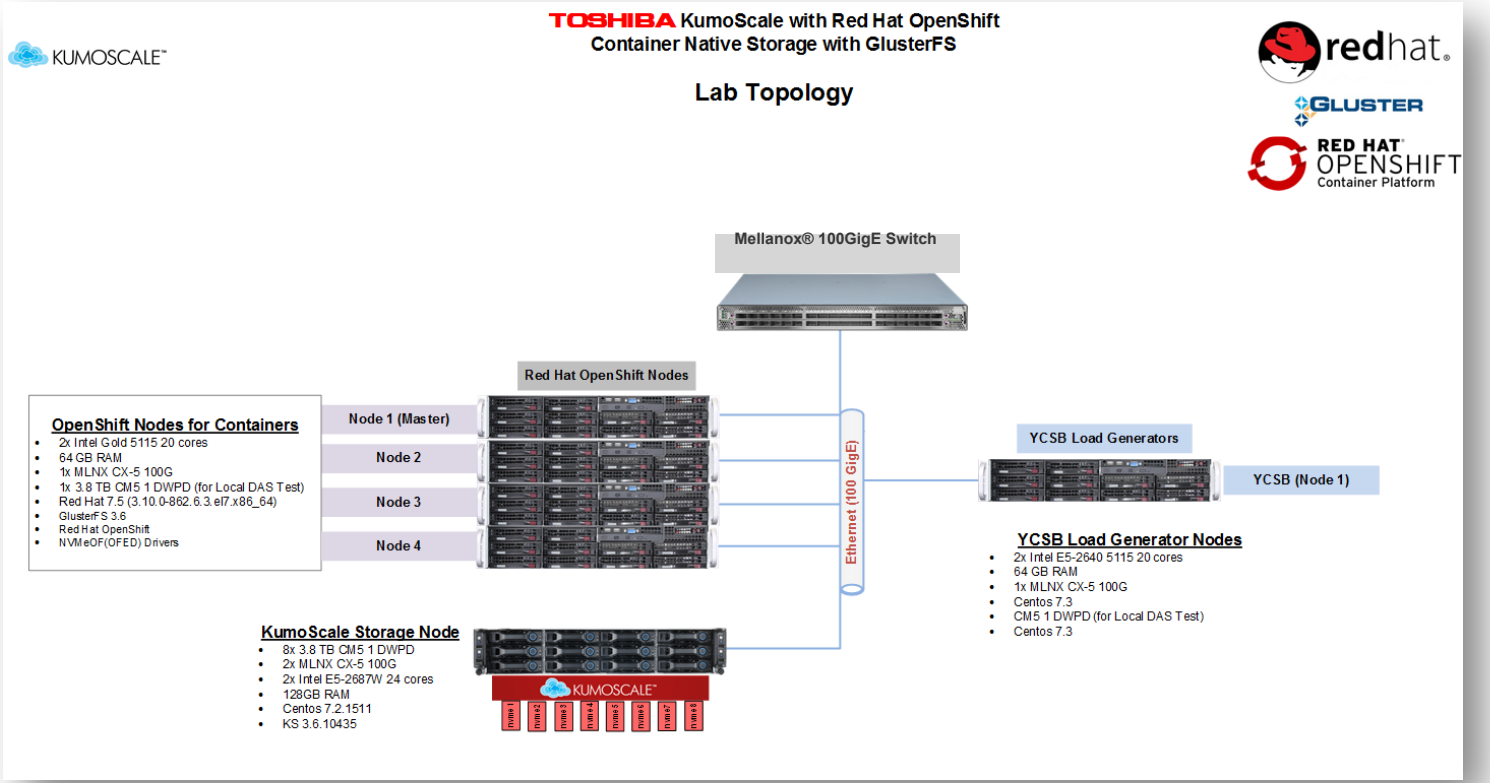
- Manages scheduling and orchestration
- Container replication and restart to defined policy.
- Demonstrate Failover of Compute Node without service disruption
- Load balancing
- Scale Out

Why Openshift? – Integrated Container Platform

RED HAT OPENSHIFT Container Platform = Linux + docker + kubernetes + Gluster SDS + Open vSwitch SDN + Mgmt. GUI

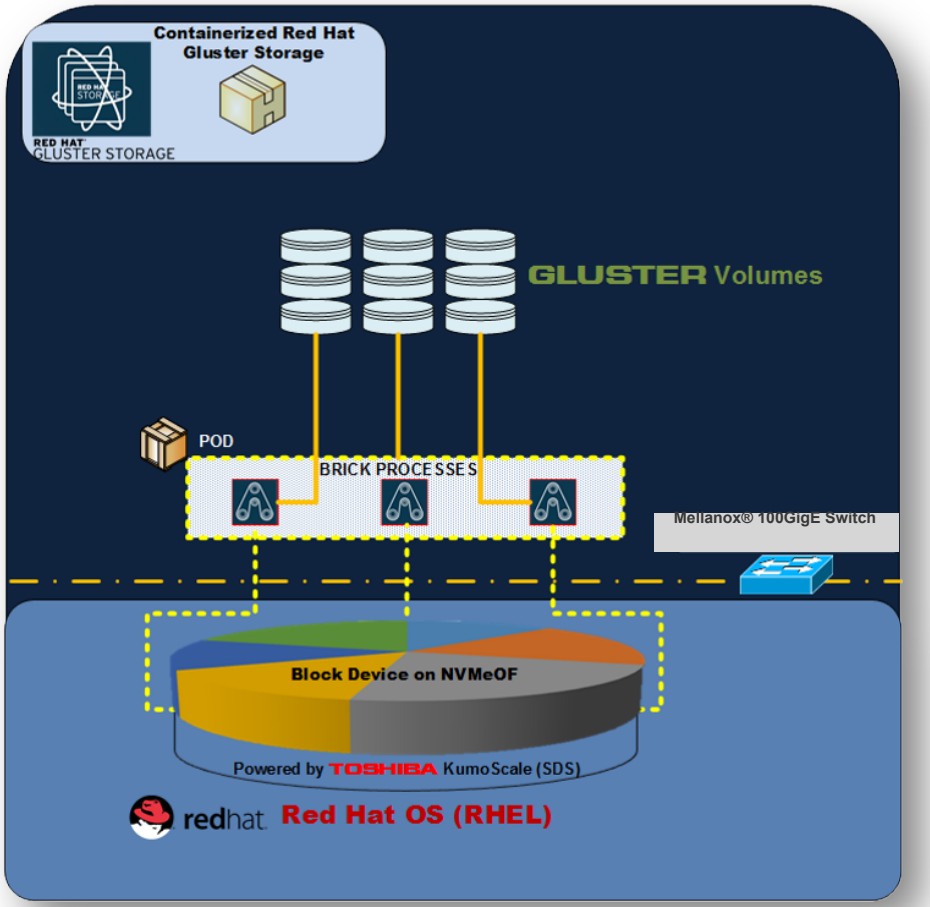
TOSHIBA
Leading Innovation

Storage Disaggregation with Container Native Storage

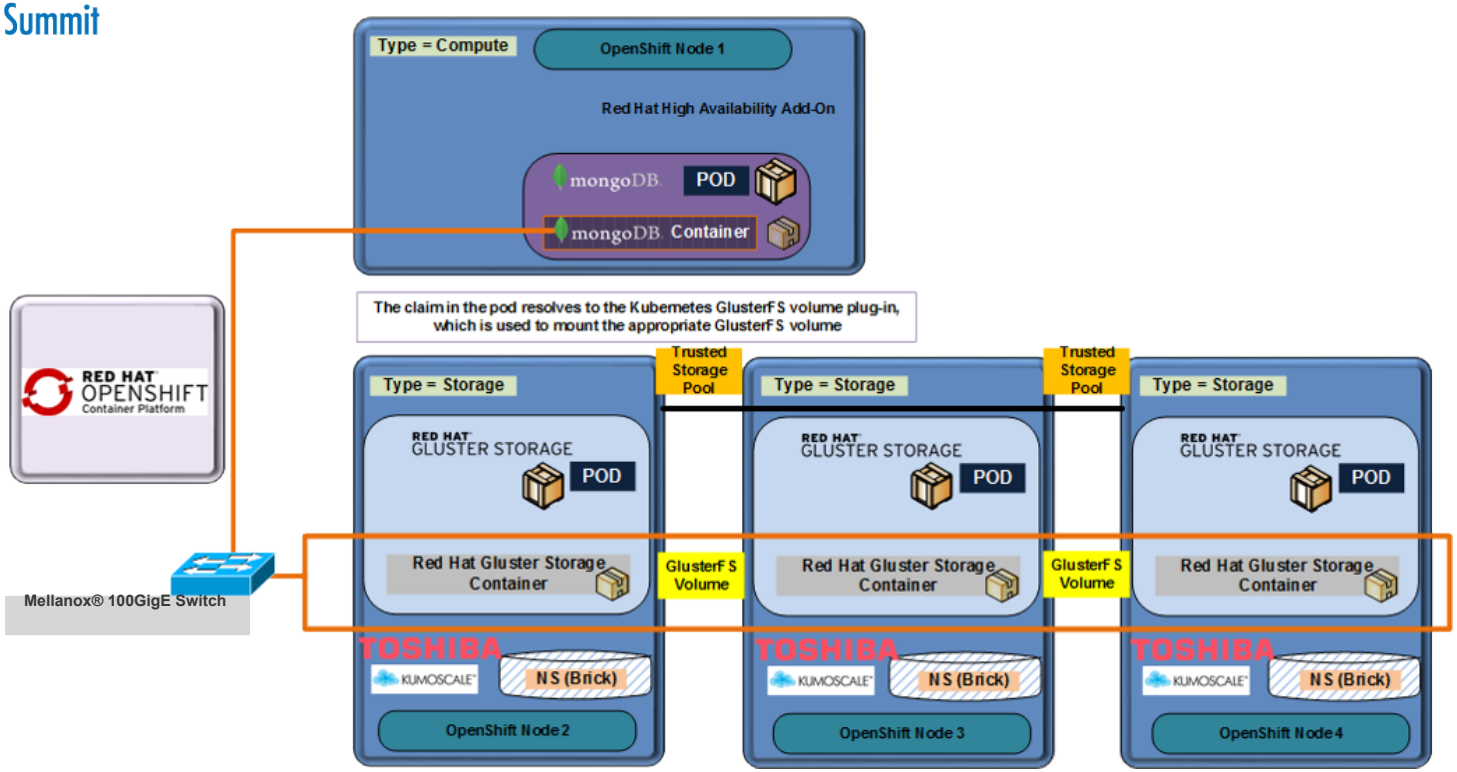


Operating System: RED HAT® Enterprise Linux 7.5
Container Platform: RED HAT® OpenShift®
File System: GlusterFS®

Storage Disaggregation with Container Native Storage with GlusterFS



Storage Disaggregation with Container Native Storage with GlusterFS



Backup Slides

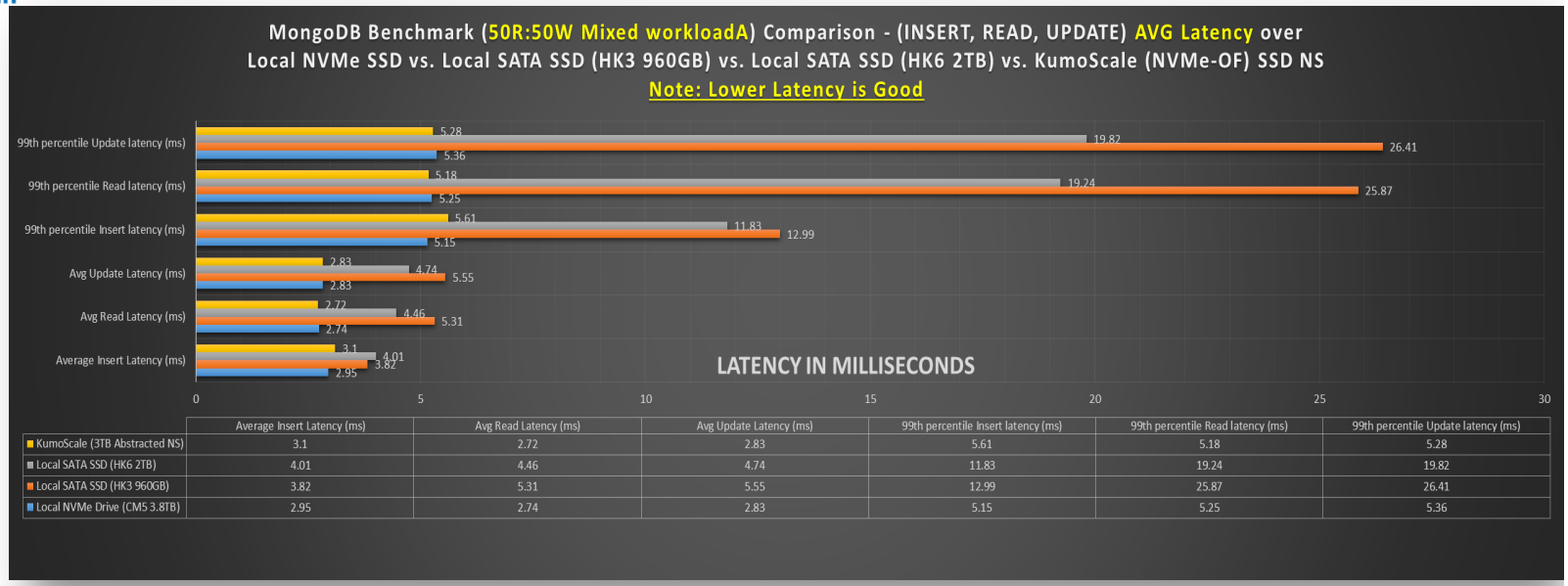
For reference only

TOSHIBA
Leading Innovation

Clustered MongoDB™ YCSB Results: Test 1 - 50r:50w Mixed workloadA

Metrics	Local NVMe (CM5 3.8T)	KumoScale NVMe-oF (3T Abstracted NS)	Local SATA SSD (HK3 960GB)	Local SATA SSD (HK6 2TB)
Configuration	3 MongoDB Shards : 1 MongoS router : 1 YCSB Engine			
Load Phase: Insert	300 Million records : 1K record size : 200 parallel Threads			
• Average Insert Latency (ms) (lower is good)	2.95	3.10	3.82	4.01
• 99th percentile Insert latency (ms)	5.15	5.61	12.99	11.83
• Insert Speed (operations/sec) (higher is good)	67,605	64,277	52,104	53,354
Run Phase: Read & Update				
• Avg Read Latency (ms)	2.74	2.72	5.31	4.46
• Avg Update Latency (ms)	2.83	2.83	5.55	4.74
• 99th percentile Read latency (ms)	5.25	5.18	25.87	19.24
• 99th percentile Update latency (ms)	5.36	5.28	26.41	19.82
• Read & Update speed (operations/ sec) (higher is good)	71,768	71,868	36,751	43,404

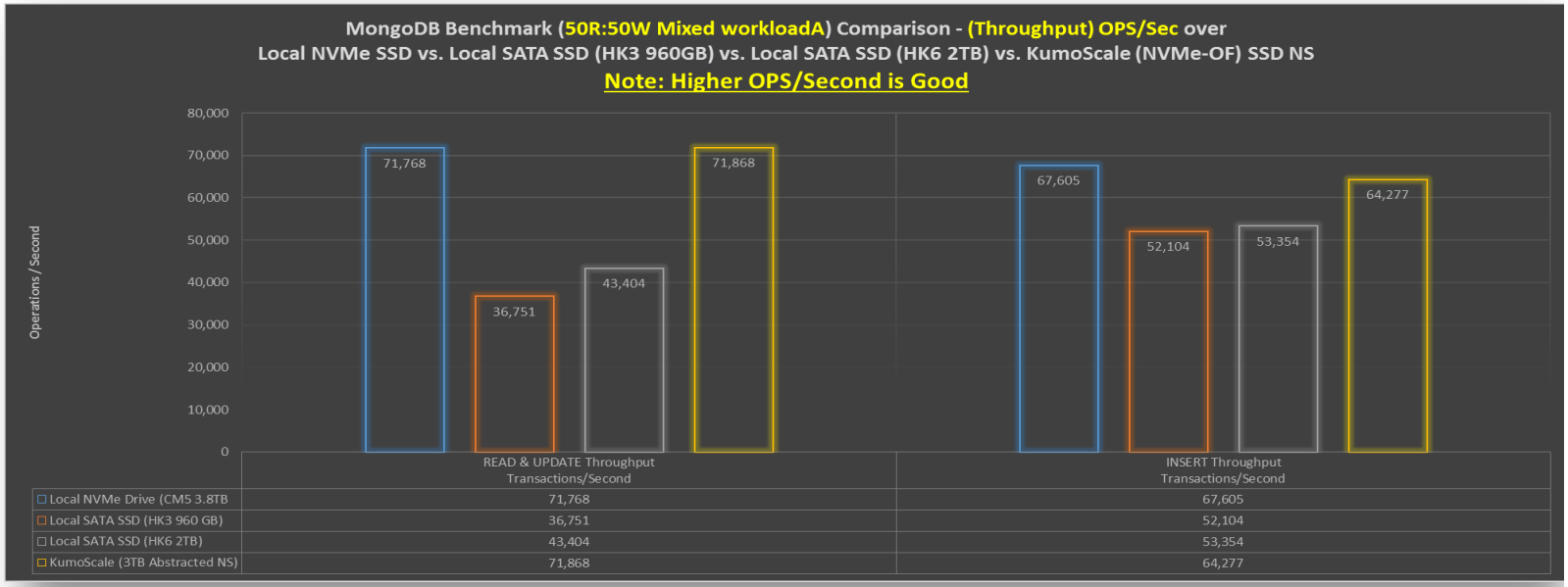
MongoDB™ Benchmark (50R:50W Mixed workloadA) Comparison – (INSERT, READ, UPDATE) AVG Latency



	Average Insert Latency (ms)	Avg Read Latency (ms)	Avg Update Latency (ms)	99th percentile Insert latency (ms)	99th percentile Read latency (ms)	99th percentile Update latency (ms)
KumoScale (3TB Abstracted NS)	3.1	2.72	2.83	5.61	5.18	5.28
Local SATA SSD (HK6 2TB)	4.01	4.46	4.74	11.83	19.24	19.82
Local SATA SSD (HK3 960GB)	3.82	5.31	5.55	12.99	25.87	26.41
Local NVMe Drive (CM5 3.8TB)	2.95	2.74	2.83	5.15	5.25	5.36

The above chart illustrates the AVG Latency Operations over the given Local Target(s) vs. KumoScale (NVMe-OF) using Mellanox 100GigE Switches. KumoScale Latency shows overall better performance in comparison with others.

MongoDB™ Benchmark (50R:50W Mixed workloadA) Comparison – (INSERT, READ, UPDATE) OPS/Second → Throughput



The above chart illustrates the Operations / Second over the given Local Target(s) vs. KumoScale (NVMe-OF) using Mellanox 100GigE Switches. KumoScale Throughput shows overall better performance in comparison with others.

Q & A