High-performance persistent storage for virtualized workloads

Highlights

Deploy highly available Red Hat OpenShift Container Storage for virtualized workloads.

Enjoy scalable I/O performance with increased storage access parallelism.

Clone and migrate VMs with Red Hat OpenShift Virtualization, backed by reliable OpenShift Container Storage.

Survive storage failures with standard data replication.

Increase database performance with storage that performs similarly to unreplicated offerings.

Table of contents

Executive summary

Supporting virtualized workloads in Red Hat® OpenShift® requires reliable persistent storage that offers virtual machine (VM) live migration, provides resilience for business continuity, and yields high performance for common VM-based workloads—even under failure conditions. To understand these factors, Red Hat engineers evaluated Red Hat OpenShift Virtualization—the container-native virtualization component of Red Hat OpenShift—comparing Red Hat OpenShift Container Storage against a competitive container-native persistent storage alternative. Testing included failover, VM cloning and booting, database workloads, and simulated input/output (I/O) operations. When coupled with OpenShift Virtualization, OpenShift Container Storage demonstrated much better performance and data availability for high-concurrency virtualized workloads compared to competing container-native storage alternatives.

Red Hat OpenShift Virtualization on Red Hat OpenShift Container Storage

With OpenShift Virtualization, teams can modernize and accelerate application delivery by migrating traditional virtualized workloads directly into container-based workflows within Red Hat OpenShift. Containerized VMs run side-by-side with containers and are managed as native Kubernetes objects. Running VMs as containers is often a first step to lift-and-shift migration, allowing modernization of VM-based applications into container-native environments. Traditional VM-based workloads can be added to new and existing applications with later decomposition into container-based microservices over time.

Data storage solutions play a vital role in this process. To adequately support OpenShift Virtualization, storage solutions must provide:

- **Support for VM live migration.** VMs must be able to move between hypervisors, whether in response to failures, or for scheduled maintenance events. VM live migration lets VMs move to unaffected hypervisors without downtime or user awareness. With containers, support for read-write-many (RWX) persistent volumes (PVs) is essential, since both source and destination hypervisors must mount PVs concurrently. OpenShift Container Storage is one of the only supported products for OpenShift Virtualization that enables VM live migration with block storage—ideal for workloads like databases that require high-performance storage.
- **Data resilience.** Data is essential to modern organizations, and no organization can afford to lose data—even during failure events. Robust data resilience must be built into any storage system used to support OpenShift Virtualization. Data storage systems must react quickly to events and incorporate methods to ensure that data is safe no matter what happens. OpenShift Container Storage provides default 3x replication to make sure that data remains safe.
- **High performance for common workloads.** As traditional virtualized applications move to containers, performance remains an essential concern—particularly when VMs host databases or other centralized infrastructure. Storage is a significant factor in performance, and application and operations teams must be assured that performance can scale as future needs change. OpenShift Container Storage has a proven track record of scaling to multiple petabytes while providing high performance for databases and other data-intensive workloads.[1]

1 *For more information, read the Red Hat document on Performance and resilience for PostgreSQL.*

Red Hat OpenShift Virtualization

While most new development is shifting to containers and serverless technology, many organizations still have substantial investments in virtualized applications. In many cases, virtualized applications serve critical existing workloads—or provide vital services to new and existing containerized applications. To bridge this gap, Red Hat OpenShift Virtualization (a feature of Red Hat OpenShift Container Platform) lets developers bring VMs into containerized workflows by running a virtual machine within a container. With this approach, virtual machines can be developed, managed, and deployed side-by-side with containers and serverless, all in one platform. OpenShift Virtualization allows organizations to take advantage of the simplicity and speed of containers and Kubernetes while still benefiting from the applications and services that have been architected for VMs.

Red Hat OpenShift Container Storage

Red Hat OpenShift Container Storage is persistent software-defined storage integrated with and optimized for Red Hat OpenShift Container Platform. It runs anywhere Red Hat OpenShift does—on-premise or in the public cloud. Built on open source technologies, such as Rook, Noobaa, and Ceph® software-defined storage, the platform offers tightly integrated persistent data services for Red Hat OpenShift running in hybrid and multicloud infrastructures, offering scalability to many petabytes and billions of objects.[2]

With OpenShift Container Storage, dynamic, stateful, and highly available container-native storage can be provisioned and deprovisioned on demand as an integral part of the Red Hat OpenShift administrator console. This integration extends to unified health and performance monitoring of VMs, as well as the storage volumes used by the VMs.

Red Hat testing: Comparing storage options

A number of storage options exist for containers, presenting different advantages and disadvantages. Table 1 lists desirable VM-centric storage features keyed to technologies evaluated in Red Hat testing. A competitive container-native storage solution was included in the comparison[3] as was a simple nonreplicated Network File System (NFS)[4] configuration as a performance baseline.[5] All three storage technologies support VM live migration, but OpenShift Container Storage is the only solution evaluated that provides all of the key features and functionality required to support OpenShift Virtualization.

---

2 *Evaluator group recently validated Red Hat Ceph Storage at 5pb serving 10 billion objects.*

3 *By policy, Red Hat does not name vendors in competitive performance comparisons.*

4 *While NFS provides good performance, running a traditional NFS server without replication for data protection is not recommended for Red Hat OpenShift Virtualization.*

5 *The use of a traditional NFS server without replication for data replication is not recommended.*

Table 1. Red Hat OpenShift Virtualization storage requirements

Feature / storage	Red Hat OpenShift Container Storage	Network File System (NFS)[3]	Competitive container-native storage solution
VM live migration	Yes	Yes	Yes
Database-ready performance	Yes	Yes	No
Red Hat OpenShift integration	Yes	No	No
Resilience against host and disk failures	Yes	No	Yes
Scale-out storage	Yes	No	Yes

Lab configuration

Testing was performed on six bare-metal servers at the Red Hat Performance and Scale Laboratory. As shown in Figure 1, the testbed was comprised of six servers connected via a 25Gb Ethernet network and configured as follows:

- One deploy host
- Two Red Hat OpenShift worker nodes
- Three Red Hat OpenShift control plane nodes

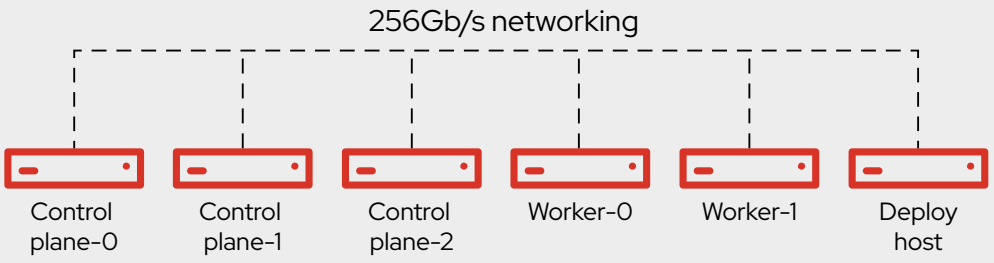

Figure 1. Laboratory configuration for Red Hat OpenShift Virtualization performance testing

Each server was configured with:

- 32 cores (Intel Xeon Gold 6130 CPU @ 2.10GHz)
- 196GB RAM
- 1x Samsung PM1725a 6.4TB NVM Express (NVMe)
- 25 Gb Ethernet networking

OpenShift Container Storage and the competitive container-native storage solution were installed one after the other, using the same NVMe disks. The NFS server was installed using one of the remaining NVMe disks. All NVMe disks in the cluster were of the same model and type.

Red Hat software installed on the testbed included:

- Red Hat OpenShift Container Platform 4.4
- Red Hat Enterprise Linux® CoreOS 44.82.202007141430-0
- Red Hat OpenShift Container Storage 4.4

Failure testing

Failures of deployed storage can interrupt critical application services. As such, storage for OpenShift Virtualization must be able to fail over gracefully, allowing applications to continue to operate. In Red Hat testing, failover was simulated by artificially failing one NVMe drive in one of the three servers. This failure caused the same issues for the two storage systems under test, removing a third of the capacity and potentially reducing performance.

- **Red Hat OpenShift Container Storage.** When evaluating OpenShift Container Storage with one NVMe disabled, performance degraded only slightly. VMs continued to operate unimpeded. Moreover, once the NVMe device was brought back online, new data was copied to the device without any user interaction.
- **Competitive container-native solution.** The competitive container-native storage solution was not able to continue operating with a failed NVMe device. With one of the three drives artificially failed, the software denied any creation of a 3x replicated persistent volume, stating that it needed the third disk to be present.
- **NFS.** Because it was based on a single server, the NFS configuration also did not survive the failure of its only disk. A NFS-based configuration would have lost all of its data, at least temporarily.

Operational testing

Viable storage solutions must support OpenShift Virtualization in the creation and operation of VMs. Operational testing consisted of cloning and booting 10 VMs under timed conditions (Figure 2).

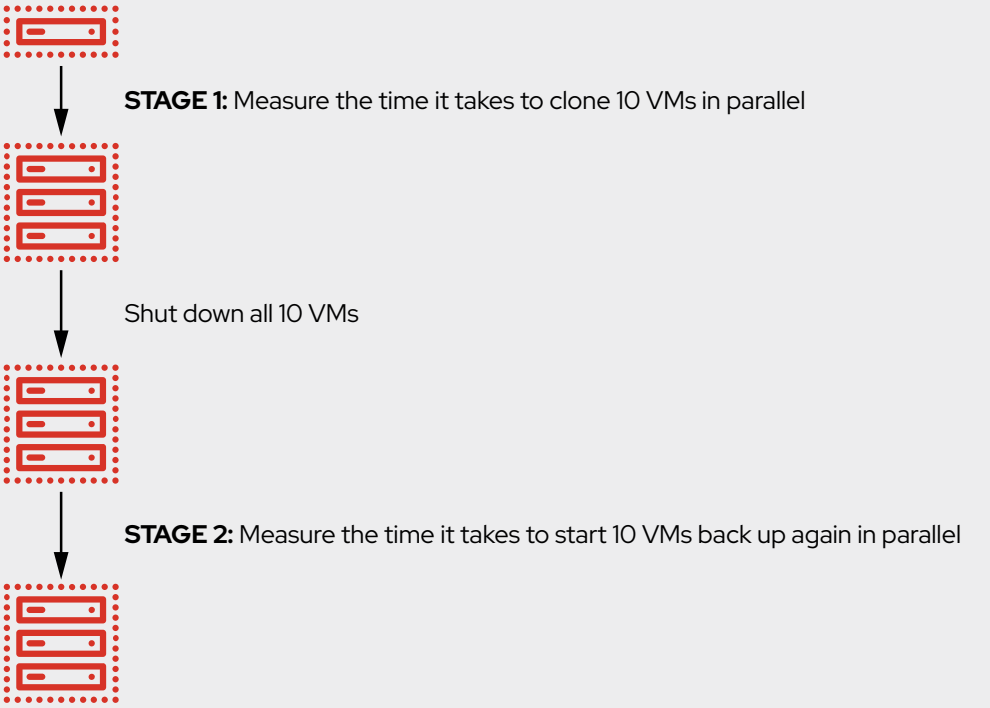

Figure 2. Staged operational testing timed the creation and booting of 10 VMs on various storage solutions

The testing consisted of two stages:

- In the **first stage**, 10 VMs were cloned from a basic Linux VM using the native DataVolume approach offered by the containerized data importer (CDI) available with OpenShift Virtualization. The Fedora® test image was 289MB in size and was imported as a DataVolume before starting the test. Time was measured between issuing the start of the cloning operation for the 10 VMs and the successful start of that VM. The source VM was not powered on during this test. CDI did not take advantage of container storage interface (CSI) snapshots during this test.
- In the **second stage**, the 10 VMs were shut down and time was measured until they all booted back up again.

Table 2 shows the operational testing results for OpenShift Container Storage, the competitive container-native storage solution, and NFS. Test results were measured in seconds (lower is better). As expected, the tests performed best on the NFSv4 baseline without replication, followed closely by the container-native storage solution. While additional processing time was not significant for OpenShift Container Storage in this test, future testing with OpenShift Container Storage version 4.6 will include the use of CSI snapshots, which is expected to improve VM cloning and boot times.

Table 2. VM cloning and booting across multiple storage solutions

	Red Hat OpenShift Container Storage (RWX block)	NFS v4	Competitive container-native storage solution
Time to clone and start 10 VMs	207.23 seconds	150.39 seconds	152.85 seconds
Time to start 10 pre-existing VMs	34.78 seconds	23.23 seconds	26.38 seconds

Database workload testing

Databases create unique stress on storage systems given their requirement for high-speed, low-latency data access. To help understand and measure the performance of underlying storage systems, Microsoft Corporation has released two tools that evaluate storage for popular Microsoft enterprise software servers:

- SQLIOSim evaluates storage for Microsoft SQL Server.
- Jetstress evaluates storage for Microsoft Exchange Server.

Engineers used these tools on the three storage systems available to compare their performance with a real-world workload. For every test run, Windows Server 2019 Standard was configured with the graphical front end, and a separate disk was used to perform performance testing. Default settings were used on both SQLIOSim and Jetstress. SQLIOSim ran for 30 minutes and Jetstress ran for two hours.

Microsoft SQL Server

Table 3 contains the result of the SQLIOSim testing across OpenShift Container Storage, NFSv4, and the competitive container-native storage solution. The results clearly showed that the container-native storage solution would not be sufficient to support good Microsoft SQL Server performance, even for the basic tests we performed.

Table 2. VM cloning and booting across multiple storage solutions

	Red Hat OpenShift Container Storage	NFSv4	Competitive container-native storage solution
Read count (higher is better)	365,272	661,015	103,902
Read time (lower is better)	807,003	724,221	496,813
Write count (higher is better)	970,728	469,401	667,623

	Red Hat OpenShift Container Storage	NFSv4	Competitive container-native storage solution
Write time (lower is better)	11,563,258	2,655,696	556,148,133
Total I/O time (ms) (lower is better)	14,068,046	10,181,073	544,929,038
Number of times I/O throttled (lower is better)	4,215	3,258	16,224
I/O request blocks (lower is better)	143	51	6,196

As a good overview of storage performance, Figure 3 shows the number of times that the Microsoft SQL Server I/O was throttled during the test run. A lower number indicates better performance.

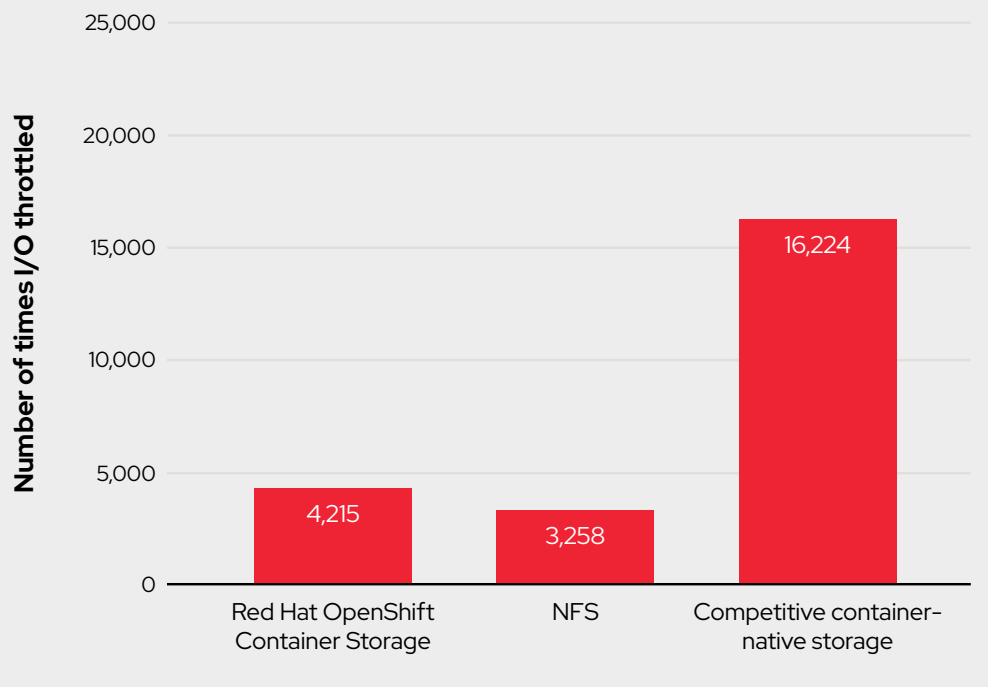

Figure 3. Number of times I/O was throttled in SQLIOSim testing (lower is better)

As expected, the unreplicated NFSv4 baseline share had the best performance with the lowest number of I/O throttling events. OpenShift Container Storage provided statistically similar performance, while delivering the additional benefit of data resiliency with 3x data replication. The competitive container-native storage solution was far behind, with almost four times the I/O throttling events compared to OpenShift Container Storage. With the combination of better performance as measured in the SQLIOSim tool, along with data replication, OpenShift Container Storage delivers significant advantages to organizations running Microsoft SQL Server.

Another indication of Microsoft SQL Server performance is the total I/O time that was observed during the SQLIOSim test. In this case, engineers had to employ a log-scale chart (Figure 4) to depict results across the three tested platforms on the same chart, as they diverged so greatly. Again, OpenShift Container Storage provided good performance, only slightly behind that of nonreplicated NFSv4, while the competitive container-native storage provided nearly 40 times longer I/O times for Microsoft SQL Server.

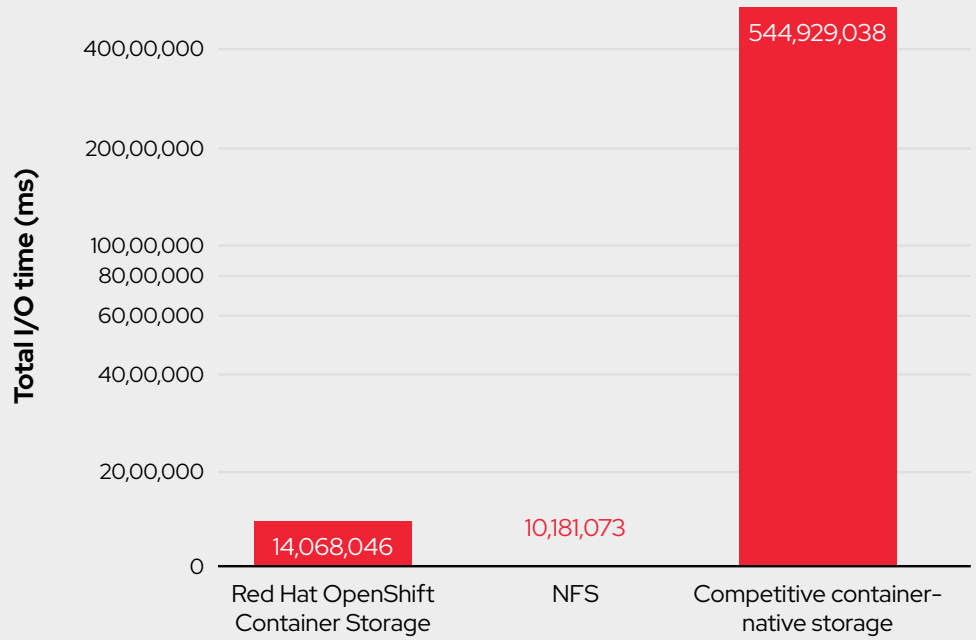

Figure 4. Total I/O time - log scale (lower is better)

Microsoft Exchange Server

Jetstress testing of Microsoft Exchange Server used mostly default settings. We instructed Jetstress to conduct testing on our three familiar storage options. Jetstress performance metrics span regular work, housekeeping, and log replication. Engineers focused on the total I/O test, which includes all of the transactions conducted during the test. Jetstress total I/O results are shown in Table 4.

Table 4. Jetstress total I/O results including log replication and housekeeping

	Red Hat OpenShift Container Storage	NFSv4	Competitive container-native storage
I/O database reads average latency (ms) (lower is better)	0.949	0.527	0.645
I/O database writes average latency (ms) (lower is better)	4.823	2.572	60.878
I/O database reads/second (higher is better)	2,452.122	2,863.631	261.386
I/O database writes/second (higher is better)	2.311.354	2829.342	178.823
I/O database reads average bytes (higher is better)	33,703.868	33,542.569	49,618.481
I/O database writes average bytes (higher is better)	33,775.417	33,930.028	36,923.293
I/O log reads average latency (ms) (lower is better)	0.488	0.353	0.429
I/O log writes average latency (ms) (lower is better)	1.685	0.634	16.148
I/O log reads/second (higher is better)	2.484	2.575	0.387

	Red Hat OpenShift Container Storage	NFSv4	Competitive container-native storage
I/O log writes/second (higher is better)	218.210	360.786	31.112
I/O log reads average bytes (higher is better)	4,096	4,096	4,096
I/O log writes average bytes (higher is better)	19,576.674	16,342.353	16,045.729

The team was mostly interested in the amount of reading and writing that can be done per second. Figure 5 shows the data for I/O database reads per second, comparing OpenShift Container Storage, NFSv4, and the competitive container-native storage solution. Again, replicated OpenShift Container Storage approached the performance of nonreplicated NFSv4, while the competing container-native storage solution generated performance that was lower by an order of magnitude. The numbers for I/O database writes per second were similar and are not shown graphically.

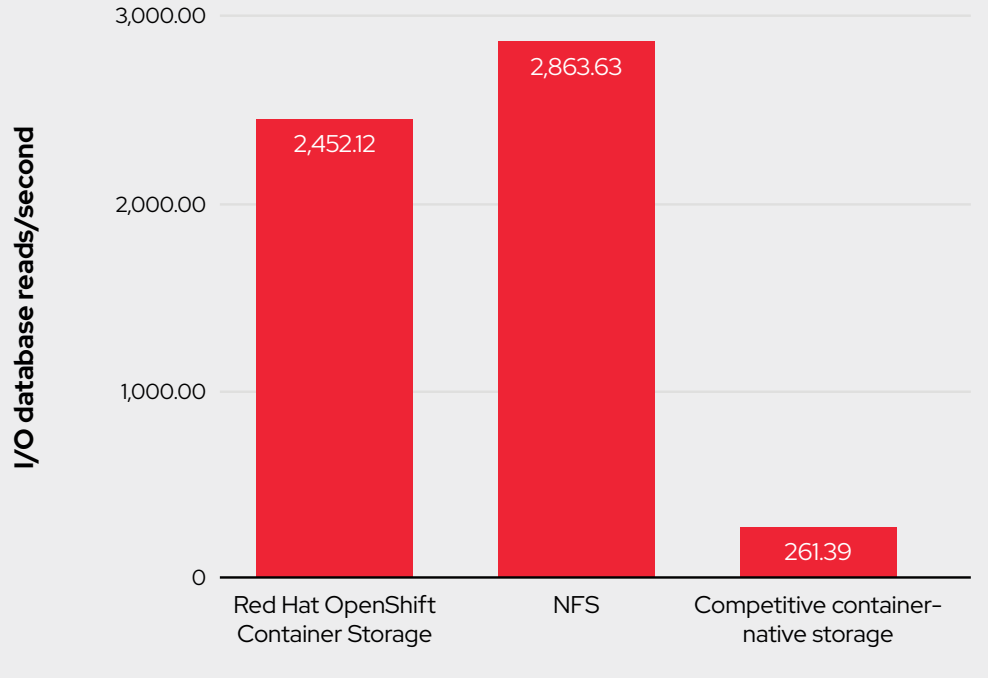

Figure 5. Jetstress I/O database reads per second (higher is better)

Simulated I/O testing

As the third test, engineers extended Ripsaw – Red Hat's internal OpenShift benchmarking tool – to run FIO workloads inside of VMs. The goal of this test was to measure storage performance for highly parallelized applications with different block sizes. Ripsaw was modified to take the existing "fio_distributed" workload that scheduled FIO servers and clients in pods, teaching Ripsaw to run the FIO servers in VMs instead. As explained, testing always ran on 10 VMs in parallel to simulate highly parallelized workloads. Test runs involved variable block sizes, though all tests resulted in similar performance. For these tests, NFSv4 was omitted and both OpenShift Container Storage 4.4 and 4.5[6] were included.

In this FIO configuration, there is a single FIO client that distributes jobs to the servers. The FIO servers are headless daemons waiting for a client to send them work that they process. After finishing their work, the FIO servers send the benchmark results back to the client for collection. In Ripsaw, the FIO client gathers the benchmark results and forwards them to ElasticSearch, where they can be analyzed by a human.

For the first set of tests, the number of VMs was fixed at 10 and the iodepth was held fixed at 4. Selecting iodepth=4 allowed gradually increasing numjobs (the number of processes) without immediately overwhelming the underlying storage. Figures 6-8 illustrate the performance of OpenShift Container Storage 4.4, 4.5, and the competitive container-native storage option while numjobs was set to 1, 10, and 20 respectively and iodepth was held constant at 4. These charts represent the average I/O operations per second (IOPS) that were achieved on 10 VMs during the test.

For storage, an OpenShift Container Storage RADOS Block Device (RBD) disk was configured in RWX configuration with 150GB of capacity. Each FIO process in the test created their own 5GB file and wrote in randrw mode. Randrw was set to issue 50% reads and 50% writes. For the competitive container-native solution, we used similar test files, but used pvcvolumemode: Filesystem since this competitive storage solution does not support RWX Block persistent volumes.

Testing showed that when there was a low degree of parallelism – e.g., a single job – the competitive container-native solution performs very well, outperforming OpenShift Container Storage 4.5 by 41%. However, most organizations run more than a few storage-consuming pods or VMs in their Red Hat OpenShift clusters. When multiple processes compete for resources, the performance of the competitive container-native solution drops considerably.

6 *At the time of this testing, OpenShift Container Storage 4.5 was a release candidate.*

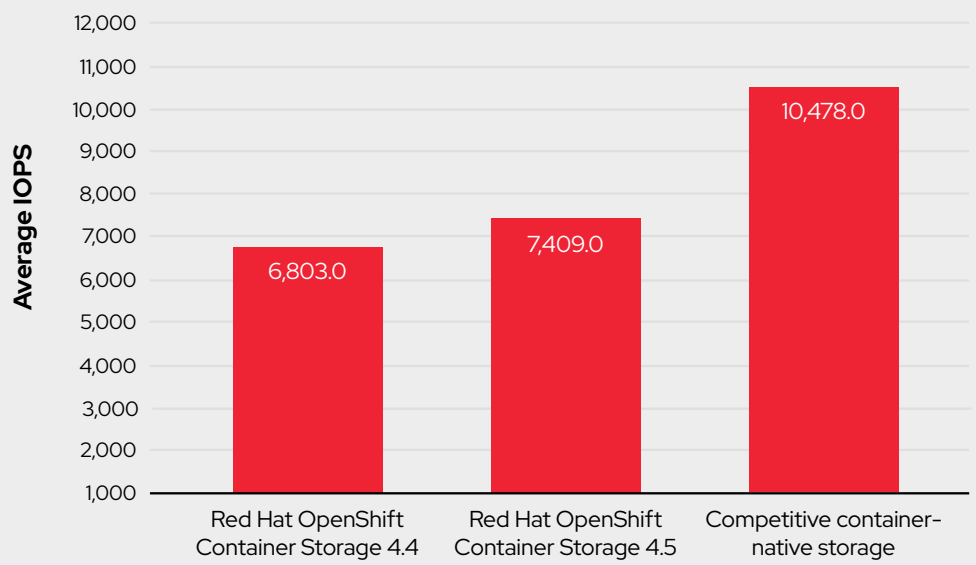

Figure 6. Average I/O operations with single-threaded, low concurrency (numjobs = 1, higher is better)

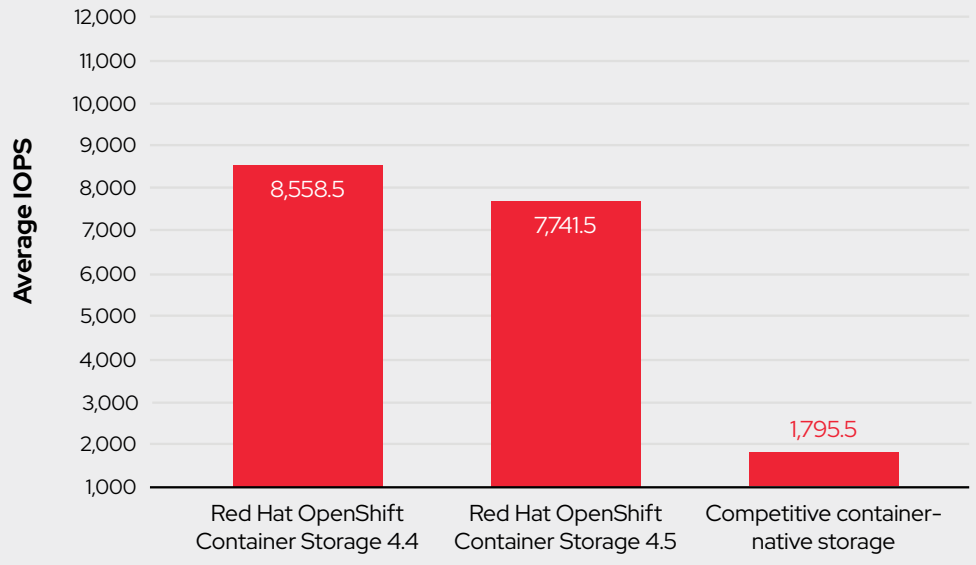

Figure 7. Average I/O operations with multithreaded concurrency (numjobs = 10, higher is better)

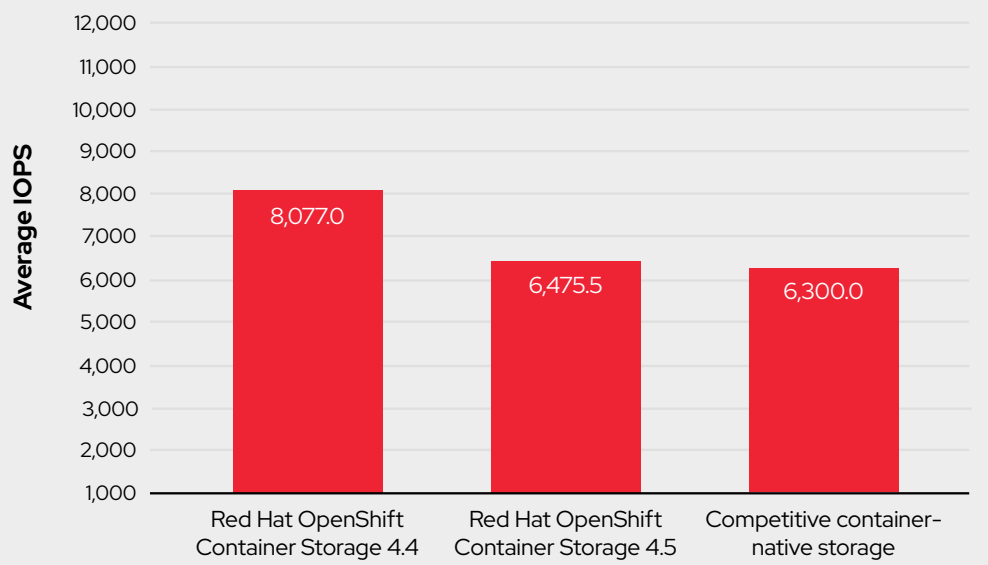

Figure 8. Average I/O operations with multithreaded, higher concurrency (numjobs = 20, higher is better)

Next, engineers left the numjobs variable set to 20, and increased iodepth from 4 to 20, and then to 40 (Figures 9-10). As numjobs increased, the parallelization of storage access became greater as well. With increased parallel access to storage, the performance of the competitive container-native solution decreased until it dropped to one-fifth of the initial result. Over the course of testing, OpenShift Container Storage 4.4 slowly decreased in performance, indicating that it was close to performance saturation. In contrast, the OpenShift Container Storage 4.5 release candidate showed increasing performance scalability with increasing parallelism.

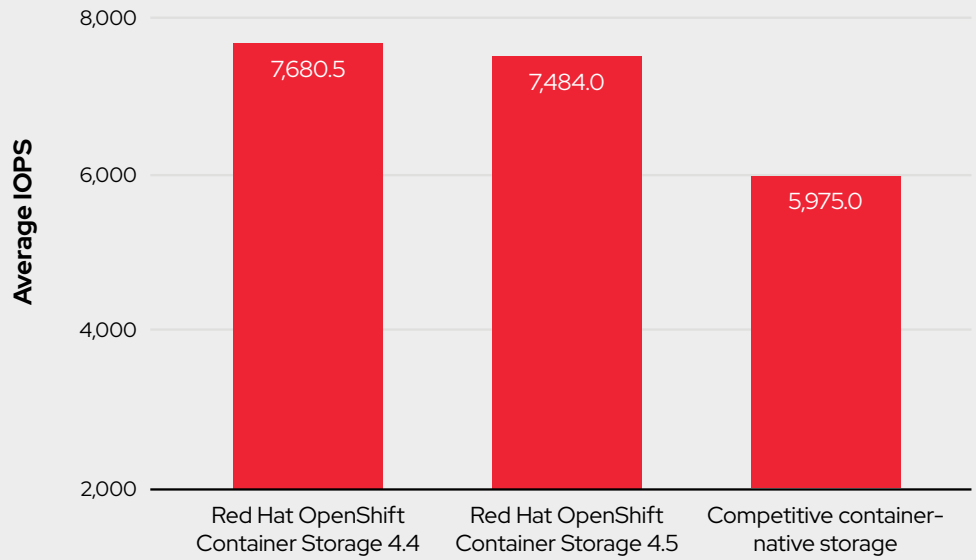

Figure 9. Average I/O operations with iodepth = 20 (higher is better)

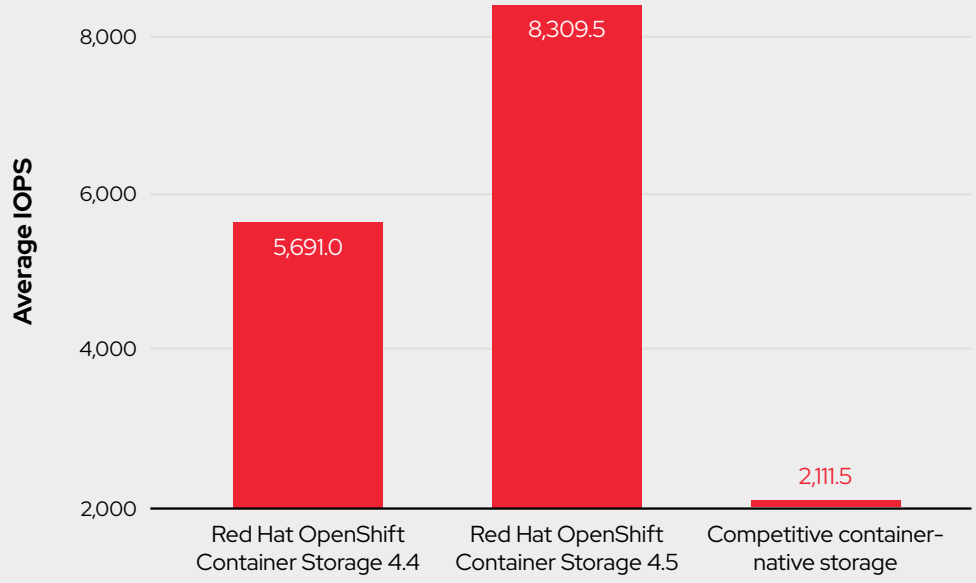

Figure 10. Average I/O operations with iodepth = 40 (higher is better)

Conclusion

Red Hat OpenShift VM workloads with a high degree of concurrency (scale) performed much better when running with Red Hat OpenShift Container Storage than when running with the competitive container-native storage solution. Red Hat OpenShift VM workload performance was generally comparable when running with OpenShift Container Storage as compared to when running with NFSv4, despite the NFSv4 configuration providing no replication nor data protection. OpenShift Container Storage provides the RWX persistent volumes that VMs require for live migration, along with the performance and redundancy needed for demanding enterprise applications.

About Red Hat

Red Hat is the world's leading provider of enterprise open source software solutions, using a community-powered approach to deliver reliable and high-performing Linux, hybrid cloud, container, and Kubernetes technologies. Red Hat helps customers integrate new and existing IT applications, develop cloud-native applications, standardize on our industry-leading operating system, and automate, secure, and manage complex environments. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500. As a strategic partner to cloud providers, system integrators, application vendors, customers, and open source communities, Red Hat can help organizations prepare for the digital future.

facebook.com/redhatinc
@RedHat
linkedin.com/company/red-hat

NORTH AMERICA
1 888 REDHAT1

EUROPE, MIDDLE EAST, AND AFRICA
00800 7334 2835
europe@redhat.com

ASIA PACIFIC
+65 6490 4200
apac@redhat.com

LATIN AMERICA
+54 11 4329 7300
info-latam@redhat.com

redhat.com
#F26641_1220