Lab Insight

Scalable Storage Performance for Cloud Native Applications

Authors: Russ Fellows &
Mohammad Rabin

June 2021

Evaluator Group

Red Hat Data Services

Executive Summary

There are two major trends occurring in IT, both of which have implications for how companies will operate and pay for their IT application needs. The first trend is the now widespread adoption of public cloud computing, along with the use of both private and public clouds, known as a hybrid cloud. The second is the rise of cloud native applications, which are built and operated using containers and Kubernetes. This shift gives companies an opportunity to examine their current storage environment and determine an optimal approach to support the new environments.

The world of containers and cloud native applications is rapidly evolving and gaining traction across a wide section of companies, regardless of their industry. Evaluator Group's recent research shows that 67% are either in production or testing to move containers into production.[1] This is due to the design of cloud native apps that enables them to run across hybrid clouds, including on premises and public cloud environments. As a result, companies that deploy cloud native apps are better able to leverage hybrid clouds to increase their IT efficiency.

With a change in how cloud native applications utilize storage, new Container Native Storage systems have emerged that are built on a cloud native infrastructure, while existing storage has been modified to support these apps are termed Container Ready Storage. Although storage performance is a critical factor to enable scalable cloud native applications, to date there have been few independent performance comparisons of Container Native Storage and older Container Ready Storage products. Although vendors do provide performance data, it is often presented in such a way that makes comparisons difficult if not impossible.

Evaluator Group was asked to compare several different storage solutions for cloud native applications and evaluate their performance, scalability, manageability and ease of use. Our testing was designed to be unbiased providing accurate comparisons. The test environment and workloads were designed to compare storage options available for cloud native applications. Testing occurred within Evaluator Group's test lab and was designed to directly compare three leading software defined storage offerings. We evaluated Red Hat OpenShift Data Foundation (ODF) storage, along with two other storage solutions.

A summary of the results of Red Hat OpenShift Data Foundation vs. competitive alternatives:

- Red Hat delivers higher performance than both competitors in all tests
    - Up to 3.0x performance using standard configuration (default CPU and RAM)
    - Up to 7.5x performance using optimized configuration (which utilized more of the available CPU and RAM per node)
- Red Hat is easier to install, monitor and manage (installed and managed from OpenShift console)
- Red Hat is highly customizable for higher performance (vs. limited customization for competitors)

---

[1] Evaluator Group – "Hybrid Cloud in the Enterprise" Research Study, to be released July 2021

Application Requirements for Hybrid Clouds

Evaluator Group works with IT organizations of various sizes across multiple countries and industries to assist with their IT strategies. Additionally, Evaluator Group performs research into new technologies and areas of focus for CIOs and IT professionals. The rise in use of cloud computing is having a significant transformational impact on IT organizations and companies of all sizes. Along with new technologies and architectures, companies are experiencing new ways to manage their on premise and public cloud deployments in a hybrid cloud environment.

IT organizations are now challenged with the need to manage new container-based applications while still supporting existing VM and physical infrastructure. Moreover, it is important that new cloud native environments can be managed by IT generalists, without the need for highly specialized Kubernetes “Dev/Ops” staff. Thus, our testing was designed to evaluate the ease of deployment and manageability aspects from an IT generalist’s perspective, rather than that of a Kubernetes, dev-ops centric operator.

In new research by Evaluator Group of over 300 IT professionals, we found that 70% of respondents are planning for container-based applications to make up all or the majority of their application workloads. Additionally, 69% of these respondents have either deployed Kubernetes in production or are testing deployments with the intention of moving to production. Shown in Figure 1 are results from an upcoming Evaluator Group research study on the question of their plans for container deployments.

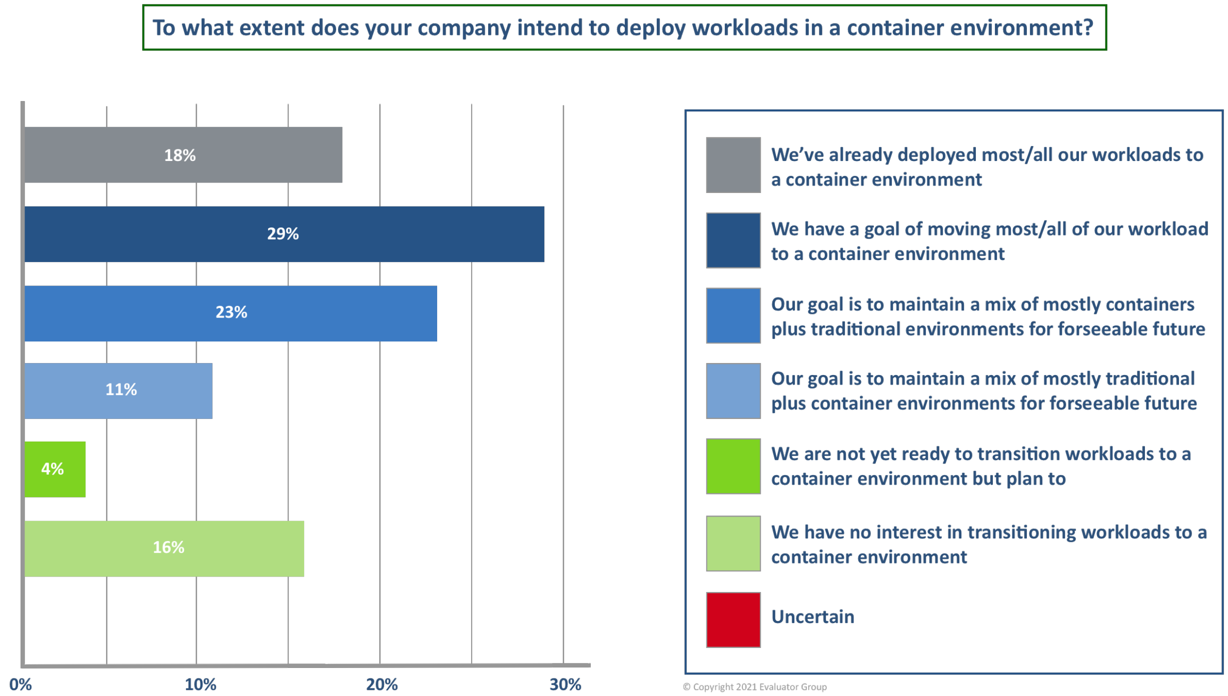

Figure 1: Container Application Survey Results (Copyright: Evaluator Group 2021)

Modern, cloud native applications require storage that is able to meet a wide range of application data requirements including good manageability and multiple application access methods, while providing scalable capacity and performance. The premise of cloud native applications is their ability to operate across IT hybrid clouds, including bare metal and virtual deployments along with support for multiple public clouds.

Desirable storage attributes for cloud native applications in hybrid clouds:

- Ease of use and manageability (Day 2 operations)
  - Easy to deploy and use for existing IT staff – Day 1 operation (NOT Kubernetes experts)
  - Ability to manage using existing tools and consoles (i.e., OpenShift)
- Support multiple application access methods (block, file and object)
- Flexible deployment options, including bare-metal and virtualized via public clouds
- Scalable performance and capacity to meet application service level requirements
- Ability to provide file recovery and near-instant application recovery

Red Hat OpenShift Data Foundation Overview

Red Hat developed one of the leading platforms for building and operating cloud native applications, known as OpenShift. Red Hat OpenShift Container Platform provides an integrated set of components necessary for running container applications utilizing Kubernetes. Red Hat OpenShift Data Foundation provides simplified access and dynamic scale, a consistent experience for using persistent storage with data services anywhere that the OpenShift Platform operates, across on-premises infrastructure and public or hybrid clouds. Deploying Red Hat OpenShift Data Foundation simplifies data management and allows on-demand storage provisioning as an integral part of orchestrated, container-based environments.

Red Hat OpenShift Data Foundation's features include:

- Container Native Storage (CNS) built to utilize and support Kubernetes / OpenShift environments
- Delivers storage performance at scale, increasing performance as application load increases
- Enables developers to consistently and effectively handle a wide range of workload types through a single unified container storage with data services platform – supporting file, block and object storage
- Simplifies application access, management and monitoring through tight integration with OpenShift Container Platform and its dashboard
- Flexibility - supporting deployments across the hybrid cloud including on-premises bare metal, or virtualized on VMware, AWS, Azure and other public cloud environments
- Provides data resiliency features that organizations require - including cloning, and snapshot copies. Coming in 2021: Disaster recovery with remote site replication capabilities.

Evaluation of Red Hat ODF vs. Alternatives

Evaluator Group tested Red Hat OpenShift Data Foundation vs. two competitors in our Boulder, Colorado labs, using Evaluator Group personnel and equipment, along with the use of software providing cloud-native storage for use by container applications running on OpenShift Container Platform. Test equipment also included the use of equipment loaned by Intel Corporation including Intel server platforms, Intel CPU's and Intel NVMe solid-state devices.

The primary objective of these tests was to fairly compare several different storage solutions for cloud native applications in terms of their performance, scalability, manageability and overall ease of use. In order to not disadvantage Container Ready Storage (CRS) that does not run under Kubernetes, we chose to setup different hardware clusters to accommodate both CNS and CRS types of storage. Additionally, the CNS storage we tested preferred 3 nodes, while the CRS storage preferred 4 nodes. Thus, in order to equalize these environments, we utilized slightly less powerful nodes for the 4-node CRS deployment so that the total storage, CPU, memory and networking was equivalent to the 3-node CNS deployment. Moreover, there were effectively two "test environments'' as shown in Figures 2 and 3.

Testing included the deployment and setup of storage along with the application environment. High level details of the environment are provided herein, with additional details in the Appendix. There are two diagrams, one for the Container Native Storage, and another for testing Container Ready Storage.

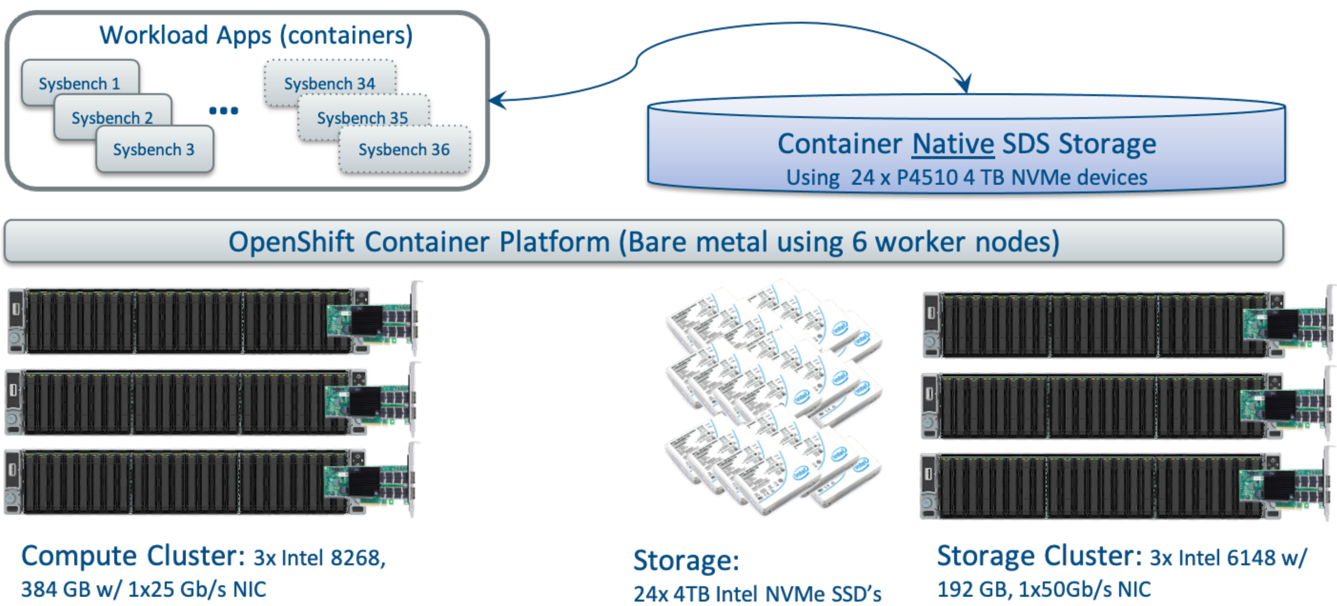

Figure 2: CNS Test Environment for ODF and Vendor B (source: Evaluator Group)

As seen in the two diagrams, the test environments were designed to be equivalent, providing no advantage to either approach of delivering storage for containerized applications.

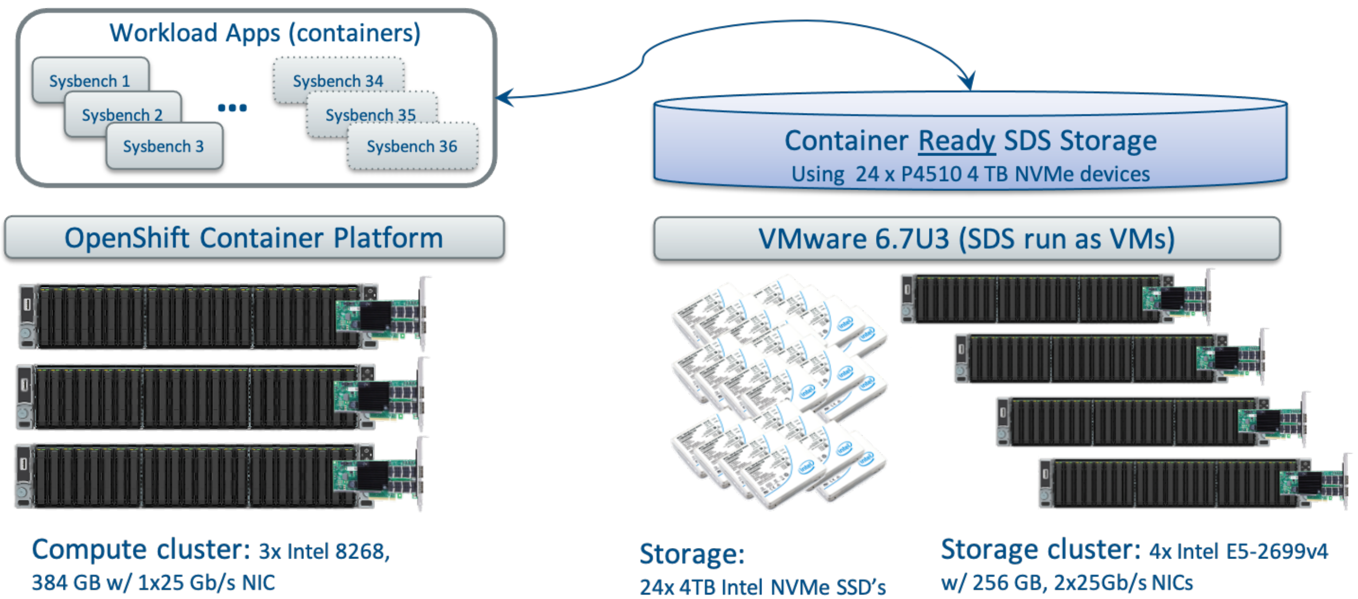

Figure 3: CRS Test Environment for Vendor A (source: Evaluator Group)

Test Overview

Evaluator Group's testing was designed to focus primarily upon quantitatively measurable results, such as performance along with measurements of system resource utilization and other measurable aspects utilizing a database application (sysbench) and a common database (PostgreSQL) for all tests. Additionally, we report upon qualitative and other aesthetic properties that are not easily measured, such as integration, ease of use and other related factors.

As shown above, there were two similar test environments used, one for container native storage with Red Hat and "Vendor B", and another for the container ready storage "Vendor A." In order to accommodate these different deployments, the application workload was purposely segregated to a logical cluster of 3 nodes running OpenShift Container Platform, even though the container native storage services and applications could be co-resident, or hyperconverged.

Another important consideration was the configuration of each of the storage solutions. Evaluator Group performed the setup and configuration of each of the three solutions multiple times, in order to ensure we understood all configuration option possibilities. We purposely chose to utilize the highest performing choices for both of the competitors, "Vendor A" and "Vendor B" in order to provide un-biased testing.

OLTP Performance Summary

The primary comparison between Red Hat OpenShift Data Foundation and two leading competitors was the maximum number of database transactions per second completed, with application instances running in OpenShift Container Platform. The test tool utilized to create the databases, fill them with data and run an application workload profile was the "sysbench" tool. More details are provided on the use and configuration of sysbench is provided in the Appendix. Additionally, orchestration of the workloads was performed by the "sherlock" tool, which created the on-line transaction processing (OLTP) workloads[2]. Used together, sherlock created container instances of sysbench databases that ran in the OpenShift / Kubernetes environment utilizing storage from each of the three tested vendor products. See Appendix for the more information on both sherlock and the sysbench tools.

It is important to note that each of Vendor's A and B's configurations were optimized to achieve the best performance possible, and specifically required several days of working with one vendor to achieve the results presented. Additionally, the data protection used was the default of 3 copies for OpenShift Data Foundation and Vendor B and Mirrored RAID in the case of Vendor A. This provided approximately 24 TB of usable capacity in all cases. For more details on the testing tools utilized see Appendices.

Performance results are shown in Figure 4 below:

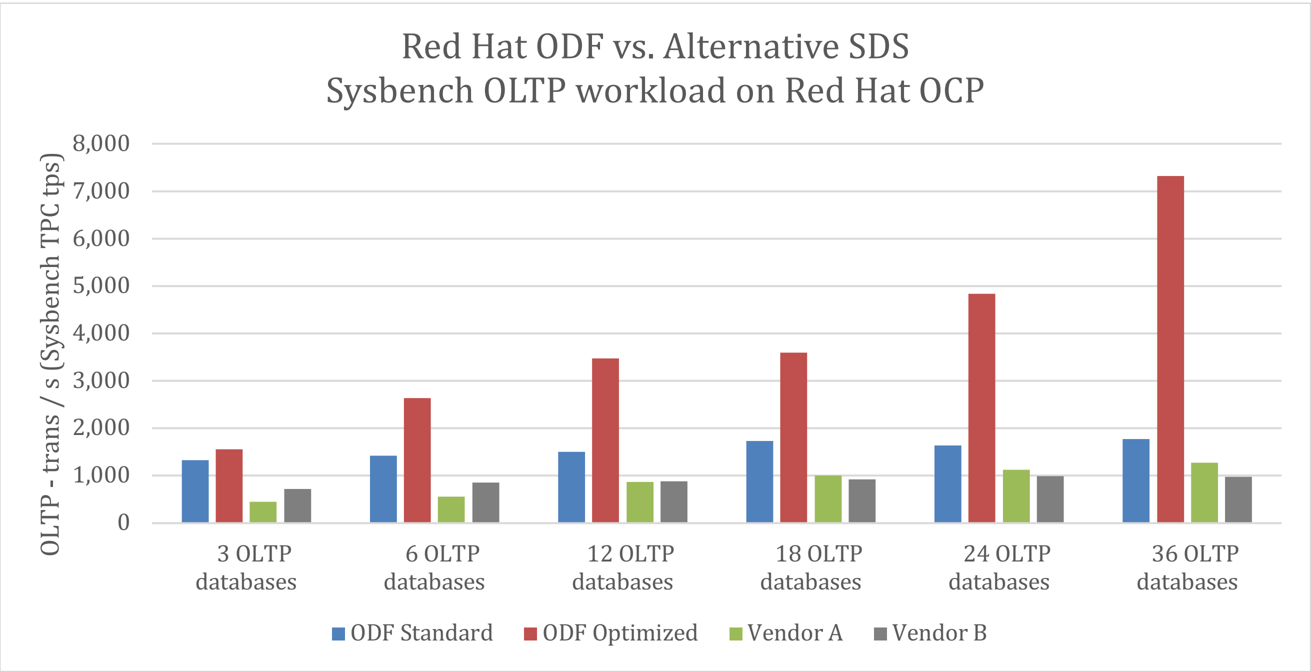

Figure 4: Red Hat ODF Storage Performance vs. Alternatives (Source Evaluator Group Testing)

The only difference between the test environments was the use of different storage and the associated CSI driver interface necessary to operate within Kubernetes and OpenShift. Each PostgreSQL application instance ran in its own container, utilizing container storage provisioned as a persistent volume, and allocated by the CSI dynamic provisioner. The tests were designed to scale-up and find the effective total transaction limit when running multiple database container instances. Testing started with 3 containers

[2] Note: this was an "OLTP, TPC-C like" workload, but was NOT the TPC-C benchmark from **www.tpc.org**

and scaled up to a total of 36 instances, with the total transactions per second measured for each type of storage. Importantly, we utilized nearly 18 TB out of 24 TB available with all 36 instances. A common mistake in many tests is using a small capacity that then gives unrealistic results, by returning data from DRAM cache. The configuration for the standard and optimized configurations of OpenShift Data Foundation are provided in the Appendix.

OLTP Performance Comparison

- Red Hat OpenShift Data Foundation outperformed the two competitors in every case tested
- Using the standard configuration for Red Hat while scaling the workload vs. the competitors:
  - Red Hat achieved 1.4x – 3.0x the total TPC transactions per second as "Vendor A"
  - Red Hat achieved 1.8x the total TPC transactions per second as "Vendor B"
- Using a custom configuration for Red Hat while scaling the workload vs. the competitors:
  - Red Hat achieved 3.5x – 5.7x the total TPC transactions per second as "Vendor A"
  - Red Hat achieved 2.1x – 7.5x the total TPC transactions per second as "Vendor B"

Initial Database Build Performance

Another aspect of performance that was captured was the time required to build and populate the 36 database instances. With each instance requiring 500 GB of storage, 36 instances required nearly 18 TB of storage capacity. The Sherlock tool invoked sysbench to create containers that then populated the database tables. The data rate and the total amount of time required to complete this process was monitored for all three storage offerings.

There were significant differences in the amount of time required, with OpenShift Data Foundation able to complete 3x faster with consistent data rates. Note, shorter completion time is better and area under curves is identical, by definition, since the amount of data is equivalent. See Figure 5 below.

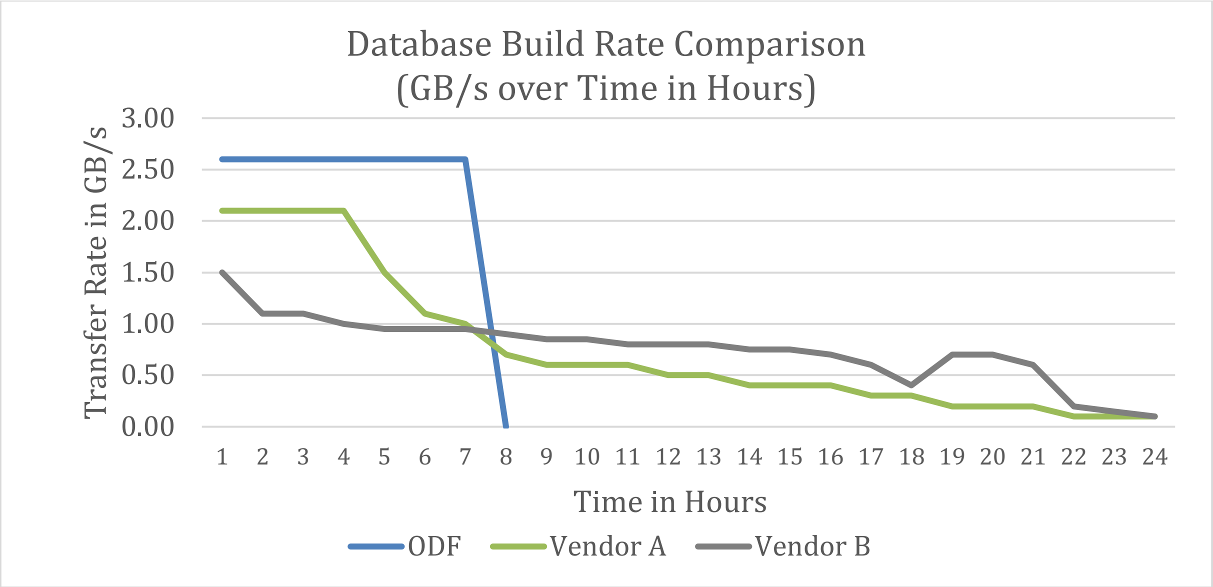

Figure 5: Database Build Time ODF vs. Alternatives (Source Evaluator Group Testing)

Database Build Performance

- Red Hat OpenShift Data Foundation provided consistently high performance that exceeded both competitors
  - Total time to build 18 TB of database data was 7 hours and 05 minutes
  - Consistent performance, data rate of approximately 2.6 GB/s
- Vendors A and B both required approximately 24 hours to create 18 TB of data
  - Total time to build 18 TB of data was almost exactly 24 hours
  - Data rate average of 750 MB/s for both vendors
  - Both vendors had good initial performance which declined over time, see Figure 5

Qualitative Test Considerations

Ease of Use

There are many factors that go into "Ease of use", which can also be highly subjective. In order to make the evaluation fair and less open to opinions, we evaluated "Ease of Use" according to several factors including: 1) Integration with Kubernetes (K8s) and OpenShift, 2) Ease of deployment and management, and 3) Ability to monitor and manage storage specific features from the provided tools.

The important factors are outlined below, with a comparison provided in Table 1.

Integration with Kubernetes and OpenShift

In order to operate with Kubernetes, a certain level of integration is required for all storage in order to be usable. The minimum practical requirement means that the storage solution supports the standard Container Storage Interface driver (aka CSI). We evaluated more than basic CSI support, looking at how easy it was to operate the storage from Kubernetes generally, but OpenShift more specifically. See Appendix for more information on CSI drivers for Kubernetes / OpenShift.

Ease of Deployment and Management

Deployment (typically included with Day 1 operations) and ongoing management tasks (Day 2 operations) are all important considerations. Although deployment may be performed infrequently, it is still an important consideration when determining how quickly capabilities can be deployed. Ongoing management is an important consideration for determining IT training and operational costs.

Storage Management Capabilities

Storage management capabilities are a basic requirement for any storage system, however; some vendors have far better integration of storage analytics into the Kubernetes / OpenShift environment than others. Two factors for consideration were the amount of analytic data provided (IO rates, throughput, latency, etc.) along with the integration of the UI monitoring into the management framework.

A Summary of Red Hat OpenShift Data Foundation (ODF) vs. the two tested competitors:

Requirement	Red Hat ODF	Competitor A	Competitor B
Ease of Deployment	Very Good: Install to OpenShift via operator or custom YAML.	Very Good: Install into VMware via proprietary tools.	Good: Support K8s and OpenShift after using proprietary installation tools
Integration with leading K8s platform	Very Good: Full integration with OpenShift management console	Limited: no UI integration with OpenShift console	Limited: no UI integration with OpenShift console
Storage Management	Very Good: Limited storage UI, full OpenShift integration	Very Good: Fully functional separate storage UI, limited K8s integration	Good: Limited separate storage UI, with some K8s integration.
Support Multiple Access Methods	Very Good: Support for raw and filesystem in containers, also Object access, no native NFS	Very Good: Support for raw and filesystem via operator in containers; also NFS and Object	Limited: Support for filesystem access only in containers, no NFS or Object access
Scalable Performance and Capacity	Excellent: Use standard or optimized configs. Performance grew 470% as workload increased 12x with optimized configuration	Good: Performance was acceptable and grew 185% as workload increased 12x using largest configuration	Limited: Performance did not scale, only grew 35% with 12x workload increase using optimal configuration
Storage Mgmt Capabilities	Good: Basic storage stats available via integrated UI	Very Good: Detailed storage stats available using separate UI	Limited: Basic storage stats available from separate UI

Table 1: Comparison of Storage Features for Hybrid Cloud (source: Evaluator Group)

Final Thoughts

Red Hat Open Shift Data Foundation should be a top consideration for companies using OpenShift Container Platform for their cloud native applications, due to performance and manageability benefits vs leading competitors. Red Hat delivered significantly better performance, with better scalability, along with a container native architecture and integrated manageability that is easier to use.

To date there have been few, independent performance comparisons between different Cloud Native Storage products or against other storage products. Although vendors do provide performance data it is often presented in such a way as to make comparisons difficult if not impossible. Our testing was designed to provide a direct comparison of 3 leading software defined storage products from large vendors using real world testing and methodologies.

The results of using a common workload and testing tools to find the performance of Red Hat OpenShift Data Foundation against other storage products, shows it delivers highly scalable storage performance that exceeds their competitors by significant margins. Not only does ODF provide higher performance, but the performance advantage also increases as the workload scales providing up to 5.7x the performance of CRS “Vendor A” and up to 7.5x the performance of CNS “Vendor B”.

When an infrastructure component has higher performance while using the same resources, companies are able to reduce both capital and operational costs. Moreover, increased performance allows companies to improve their efficiency, while delivering higher service levels to applications and end users. Typically, in head-to-head comparisons of similar products, performance advantages are measured in percentages, since the differences tend to be less than 2x, or 100%. With 400% - 600% better performance of its competitors, Red Hat provides clear performance, cost and TCO advantages.

In addition to performance, the benefits of using a CNS solution for cloud native applications can be significant compared to using a more traditional Cloud Ready Software Defined Storage solution. Cloud Native Storage has the potential to reduce infrastructure requirements, simplify the deployment and add the flexibility of being able to run applications and storage on the same nodes.

Finally, the manageability, integration and overall ease of use aspects are also important to companies looking for a storage solution for their cloud native applications. The fact that Red Hat’s installation and management are integrated into the Red Hat OpenShift Container Platform is another advantage for companies utilizing the OpenShift framework to build, run and operate their cloud native applications.

Appendix

Test Environment Details

The test environment utilized the following hardware / infrastructure, along with software / application workload setup described below.

Hardware and Infrastructure

- Two test environments as shown in Figures 1 and 2
- Red Hat OCP (OpenShift Container Platform) installed on CNS and application nodes
- Multiple workloads running as containers consuming 500 GB of capacity (from tested storage)

Compute Nodes for both CNS and CRS clusters (shown in left side of Figures 1 and 2)

- Quantity 3 - Intel 6148 servers with 384 GB DRAM and 1x 25 Gb Ethernet
- Red Hat OCP v 4.6 (installed on bare metal servers)

Storage Nodes for CNS cluster- ODF and Vendor B SDS (shown previously on right side of Figure 1)

- Quantity 3 - Intel 6148 servers with 192 GB DRAM and 1 x 50 Gb Ethernet
- Red Hat OCP v 4.6 (installed on bare metal servers)
- Total of 24 – Intel P4510 4TB NVMe devices (8 per node) passthrough to SDS storage
- **Note**: Data protection level of 3 copies of data utilized for both ODF and Vendor B

Storage Nodes for CRS cluster – Vendor A SDS (shown previously on right side of Figure 2)

- Quantity 4 - Intel E5-2699v4 servers with 256 GB DRAM and 2 x 25 Gb Ethernet
- VMware vSphere 6.7U3 (ESXi on each node with vCenter server)
- Total of 24 – Intel P4510 4TB NVMe devices (6 per node) passthrough to SDS storage
- Data protection of RAID per node, mirrored to a partner node (similar to RAID 50)
- *Note: The 3 - OCP master nodes also ran as VM's on this cluster*

Application Environment

- The open source "Sherlock" tool was used to run a Sysbench TPC workload on a PostgreSQL database, (Sherlock is available on GitHub: **https://github.com/sagyvolkov/sherlock**)
    - The database type used was "PostgreSQL"
    - The database size was set to 500 GB capacity per instance (trial and error using rows per table and number of tables parameters)
    - Storage volume (PV) was created by setting the request (PVC) to 550 GB
    - Storage access mode for PV / PVC was RWO (read write once)
    - Storage type was "filesystem" with an EXT4 mount point for PostgreSQL

  - Workload was measured on 3 separate runs, each for 15 minutes (900 seconds)
- See Sherlock configuration parameters at end of Appendix

Test Overview

- Deployment SDS Storage (measure time, steps, and ease of use):
  - Setup of storage and initial configuration including creation of K8s storage classes
  - Setup and deployment of agents and plugins (where required)
  - Setup of any remaining management or measurement tools
- Ongoing operations (measuring ability to visualize and collect metrics)
  - Setup and monitoring of VM's being protected
  - Measuring capacity utilization of entire storage pool and on a per PVC basis
- Setup and run Sysbench TPC workload (measure reported performance)
  - Utilize tool "Sherlock" for setup and running of OLTP workload with sysbench
  - Measure performance as reported by Sherlock / sysbench for OLTP workload

Test Steps

- Using Sherlock, deploy Sysbench pods using configuration file (shown below)
- Initialize storage allocated by Sherlock for Sysbench pods
- Run Sysbench PostgreSQL workload using Sherlock

The contents of the configuration file used for Sherlock are shown below. Note, this configuration will create and run a total of 36 container instances (12 databases per worker * 3 workers = 36).

Sherlock Configuration File Contents:

```
readonly KUBE_CMD=oc
readonly WORKERS_LIST_FILE=/home/kubeadmin/sherlock/Databases/workers
readonly SDS_LIST_FILE=/home/kubeadmin/sherlock/Databases/sds
readonly NUMBER_OF_WORKERS=3
readonly DB_PER_WORKER=12
readonly DB_POD_MEM=16Gi
readonly PVC_SIZE=550Gi
readonly DB_POD_CPU=5
readonly WORKLOAD_RUNTIME=900
readonly THREADS=12
readonly CLIENTS=2
readonly OUTPUT_INTERVAL=10
readonly SYSBENCH_NUMBER_OF_TABLES=400
readonly SYSBENCH_ROWS_IN_TABLE=5000000
readonly SYSBENCH_NUMBER_OF_INSERTS=1
readonly SYSBENCH_NUMBER_OF_UPDATES=1
readonly SYSBENCH_NUMBER_OF_NON_INDEX_UPDATES=1
readonly SYSBENCH_READ_ONLY=off
readonly SYSBENCH_WRITE_ONLY=off
readonly BENCHMARK_POD_MEM_LIMIT=24Gi
readonly BENCHMARK_POD_CPU_LIMIT=6
readonly NAMESPACE_NAME=postgresql-ocs
readonly DB_TYPE=postgresql
readonly DB_POD_PREFIX=postgresql-ocs
readonly DB_PVC_PREFIX=postgresql-ocs-pvc
```

```
readonly DB_NAME=redhat
readonly DB_USERNAME=redhat
readonly DB_PASSWORD=redhat
readonly STATS=true
readonly STATS_INTERVAL=10
readonly SDS_DEVICES="nvme0n1"
readonly SDS_NETWORK_INTERFACES="eno1"
readonly RBD_STATS=false
readonly DEBUG=false
readonly SDS_NODE_TAINTED=false
readonly STORAGE_CLASS=ocs-storagecluster-ceph-rbd
```

ODF YAML contents for "Default" and "Optimized" performance:

```
apiVersion: ocs.openshift.io/v1
kind: StorageCluster
metadata:
  name: ocs-storagecluster
  namespace: openshift-storage
spec:
  manageNodes: false
  monDataDirHostPath: /var/lib/rook
  managedResources:
    cephBlockPools:
      reconcileStrategy: manage
    cephFilesystems:
      reconcileStrategy: ignore
    cephObjectStoreUsers:
      reconcileStrategy: ignore
    cephObjectStores:
      reconcileStrategy: ignore
    snapshotClasses:
      reconcileStrategy: manage
    storageClasses:
      reconcileStrategy: manage
  multiCloudGateway:
    reconcileStrategy: ignore
  storageDeviceSets:
  - count: 8 # This may also be set to 1 and then expand the cluster from the UI to use all devices
    dataPVCTemplate:
      spec:
        accessModes:
        - ReadWriteOnce
        resources:
          requests:
            storage: "3726Gi"
        storageClassName: nvme-sc
        volumeMode: Block
    name: ocs-deviceset
    placement: {}
    portable: false
    replica: 3
    resources:
      limits:
        cpu: "4" #Default value is 2
        memory: "10Gi" #Default value is 5Gi
      requests:
        cpu: "4" #Default value is 2
        memory: "10Gi" #Default value is 5Gi
```

About Evaluator Group

*Evaluator Group Inc. is dedicated to helping **IT professionals** and vendors create and implement strategies that make the most value of their storage and digital information. Evaluator Group services deliver **in-depth, unbiased analysis** on storage architectures, infrastructures, and management for IT professionals. Since 1997 Evaluator Group has provided services for thousands of end-users and vendor professionals through product and market evaluations, competitive analysis, and **education**. **www.evaluatorgroup.com** Follow us on Twitter @evaluator_group*

This document was developed with Red Hat funding. Although the document may utilize publicly available material from various vendors, including Red Hat and others, it does not necessarily reflect such vendors' positions on the issues addressed in this document.