Lab Insight

Massively Scalable Cloud Storage for Cloud Native Applications

Authors: Russ Fellows & Mohammad Rabin

September 2020

Executive Summary

Cloud computing has become prevalent across companies of all sizes, and for some companies this is how all new application workloads are being deployed. With cloud computing, new applications have arisen that are designed to run independently of specific infrastructure platforms, using cloud native resources and interfaces. These "cloud native applications" are designed to operate in hybrid cloud environments, without being limited to local file structures or access requirements.

Many of these cloud native applications utilize object storage to deliver persistent data for containerized applications, rather than using file systems or other data access methods. Often, the size and scope of these applications requires the ability to scale into the billions of objects. Access to object data occurs via the S3 protocol regardless of deployment throughout a hybrid cloud environment.

OpenShift is a key component in Red Hat's portfolio of products designed for cloud native applications. It is built on top of Kubernetes, along with numerous other open source components, to deliver a consistent developer and operator platform that can run across a hybrid environment and scale to meet the demands of enterprises. Ceph open source storage technology is utilized by Red Hat to provide a data plane for Red Hat's OpenShift environment.

The 10 Billion Object Test Challenge

Evaluator Group worked with Red Hat to demonstrate the scale and performance of Red Hat Ceph 4.1, in a "10 Billion Object Challenge." Previously, Red Hat had tested a 1 Billion object cluster and this project was designed to test an order of magnitude higher scale.

In our Boulder, Colorado lab we built a 6 node, 4.5 PB Red Hat Ceph storage cluster using Seagate's Exos E 4U106 high-density JBOD storage enclosures, Seagate's Exos X16 16 TB SAS nearline enterprise drives for capacity, along with Intel NVMe SSDs for caching. Workloads were designed to test small object performance applicable for cloud-native applications. Additionally, we tested large object workload performance with object sizes that are often used by big data applications.

The configuration used for testing was able to support more than 10 billion objects, but perhaps, more important than this scale was the fact that object PUT and GET operations were deterministic, providing nearly linear performance as the system capacity grew to over 10 billion objects and 80% of usable capacity. A high-level summary of the performance includes:

- Small, 64 KB objects proved predictable performance
  - An average of more than 28,000 objects / sec for GET operations
  - An average of more than 17,000 objects / sec for PUT operations
- Large, 128 MB objects attained the following with nearly constant performance
  - An average of more than 11.6 GB / sec GET bandwidth
  - An average of more than 10.6 GB / sec PUT bandwidth

Cloud Native Application Requirements

As Cloud computing has become prevalent across companies, new cloud native applications often choose to utilize object storage in order to persist data, rather than using files or local storage resources which are limited in scope. By using "cloud native" storage via URL's, applications can run without restriction of needing specific local data access or structures. This better enables these applications to truly operate in hybrid cloud environments, across on-premises private and public clouds.

An organization that has become the de facto standards body for cloud computing is the cloud native Computing Foundation (CNCF). The CNCF identifies multiple elements for cloud native environments[1] that enable organizations to build and run scalable applications in multiple environments including public, private, and hybrid clouds. Cloud native applications utilize technologies such as containers and application interfaces to speed development, deployment, and integration.

Cloud native environments, regardless of where they are deployed, still need to provide three primary components for application run time including compute, networking and storage. Stateful applications in particular require persistent storage, and while there are many ways to provide storage, a true cloud native Storage solution should encompass several attributes listed below.

Cloud Native Storage Attributes:

- Provides storage quality of service to applications
- Delivers independence from particular infrastructure or cloud service deployment
- Provides programmatic tools for composing, deploying and maintenance
- Matches an application's persistence, availability, reliability and scalability requirements

Object Storage Uses

Object storage has been in use for several decades but has risen to prominence in the last few years. Originally object storage was designed as a means of providing additional metadata and context to storage without the overhead of a filesystem. Object storage was envisioned as a potential evolution for traditional block and file storage access, providing characteristics that could facilitate large scale with high performance multiple data types. One of the most common uses of object storage is as an object archive or repository. But more recently object storage has increasingly been used for cloud native apps.

Object Archive Repositories

In the past, object storage technology was used to provide low-cost, high capacity online data repositories. This includes media (photos, videos, audio, games, etc.) and other large content sharing and hosting services, as well as a repository for data archives. Filesystems inherently have scalability issues that are

[1]cloud native Computing's definition of cloud native: https://github.com/cncf/toc/blob/master/DEFINITION.md

overcome by object repositories, which provide highly scalable namespaces coupled with low-cost and high capacity. Object storage continues to be utilized for object repositories both on-premises and in public clouds.

Cloud Native Applications

In addition to the use of object storage for repositories, many modern applications designed for the cloud have come to rely upon objects for long term retention, or persistence of data. Rather than using file systems or traditional relational databases for data, these cloud native applications use object storage via the S3 application interface.

Thus, even though the backing object storage systems utilize the same protocols as object repositories, the applications and the data itself have significantly different use cases and access expectations. Typically, objects utilized by cloud native applications are smaller than for other uses, with object sizes of 64K and smaller.

Red Hat cloud native Portfolio

A crucial part of any application is how data is stored and retained in a way that meets the application and businesses availability criteria. Cloud native applications require technology that goes beyond traditional storage, to provide access to information regardless of when or where an application is deployed.

Red Hat OpenShift is arguably the most cohesive and successful cloud native environment available. OpenShift is both a development and runtime environment that provides a "cloud native" experience with support on several physical deployments. This includes all major public clouds as well as on-premises deployments on bare-metal and virtual machine environments.

Ceph storage technology has been maturing in the open source community for 15 years. One of the core areas of focus within Red Hat is Ceph storage technology, which provides a data plane for Red Hat's OpenShift environment, and for cloud native applications in general.

Red Hat OpenShift Container Platform

OpenShift is a key component in Red Hat's portfolio of products designed for cloud native applications. It is built on top of Kubernetes, along with numerous other open source components, to deliver a consistent developer and operator platform that can run across a hybrid environment and scale to meet the demands of enterprises.

- Built by Red Hat, OpenShift is a leading enterprise Kubernetes platform designed to provide a consistent foundation to deliver applications anywhere, with full-stack automated operations and optimized developer workflows.

- Red Hat OpenShift Container Platform (OCP) OCP is an enterprise container and Kubernetes development and deployment environment which supports open hybrid clouds. OCP includes a container runtime and registry, authentication and authorization, networking and monitoring, all of which are tested and supported for unified operations.
- Red Hat Open Shift Container Storage (OCS) provides persistent cloud native storage for container applications running in OCP. OCS is built upon Ceph and integrated into OCP to provide a data plane for applications running in OCP.

An overview of Red Hat's perspective of their OpenShift Platform (OCP), along with OpenShift Container Storage (OCS) is shown in Figure 1.

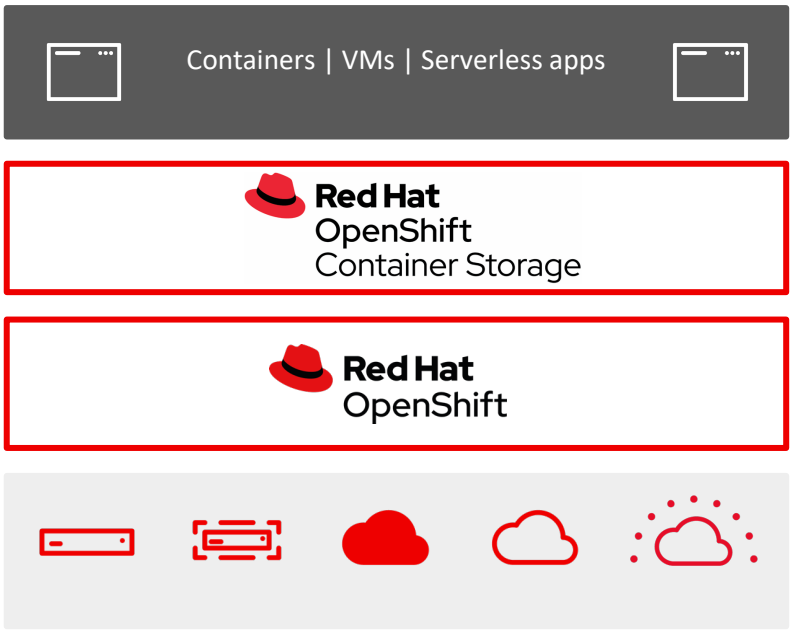

Figure 1: Red Hat OpenShift with OCS (Source: Red Hat)

OpenShift Container Storage

Red Hat's OpenShift Container Storage is based upon Ceph storage technology and is completely integrated into the OpenShift platform to provide consistent application and operational access and management of data. OpenShift utilizes key Kubernetes storage interface, the Container Storage Interface (CSI), along with dynamic provisioning of storage for applications using Kubernetes persistent volume claims (PVCs).

Red Hat Ceph Storage

Red Hat Ceph Storage is a product based upon the same underlying Ceph technology as OpenShift Container Storage. However, Red Hat Ceph Storage is independent from OpenShift, providing object storage options for environments that want enterprise class scalable object storage, but may not use OpenShift.

With its origins as an open source project, Ceph has many configuration parameters that may be used to optimize specific workloads. Ceph's concepts are logical, with the ability to scale each aspect of the system in order to meet the application workload objectives. Thus, architects may change the ratio of disk and cache capacity, speed and bandwidth to the storage devices, the amount of CPU and memory utilized for Ceph processes, and the number and bandwidth of network interfaces. Additionally, architects may choose cluster sizing and what type of data protection algorithm to use, making Ceph highly customizable. In this project we intentionally choose to go with the default tuning parameters of Red Hat Ceph Storage, so that we stay close to real production environments.

Significant architectural enhancements, including "BlueStore" have been made to Ceph to optimize both spinning and flash media types. These enhancements include management of storage devices, along with meta-data management using a key-value store along with checksums, inline compression and journaled writes via a write-ahead log.

General Architectural Recommendations

Red Hat has a number of recommendations for designing and sizing Ceph cluster configurations. Options include the number of nodes, number of storage devices per node, network connectivity, storage media types, meta-data capacity and other considerations. Red Hat has put together several recommendations, which are available via the URL in the footnote.[2]

Architecting for Large Object Size Workloads

Object storage systems have often been utilized to store relatively large objects, although what constitutes "large" vs. "small" is a somewhat nebulous distinction. Generally, anything over 1 MB in size may be considered large, with small objects being anything smaller. As with any large block or object, the goal is to achieve high data transfer rates, rather than latency and I/O rates. In order to maximize bandwidth, it is important to utilize multiple high bandwidth network links along with multiple storage connections and storage devices.

A Ceph cluster optimized for large objects would likely utilize multiple 25 Gb Ethernet interfaces per host, along with multiple SAS connected storage enclosures. Storage devices would be large, high capacity

[2] Red Hat Ceph Sizing Guidelines: https://www.Red Hat.com/cms/managed-files/st-rhcs-config-guide-technology-detail-inc0387897-201604-en.pdf

devices, along with a small amount of SSD storage utilized for meta-data. Server configurations would require high I/O and memory bandwidth and less CPU than systems designed for smaller object workloads.

Architecting for Small Object Size Workloads

A Ceph cluster optimized for small objects would be designed to maximize the number of I/O operations. Such a cluster would utilize 10 Gb/s Ethernet or higher speed connections. Storage devices should utilize performance HDD's or SSD's for higher performance, along with relatively sufficient NVMe storage to cache all meta-data. Small object server configurations would require more CPU compared to large object configurations, with PCIe and memory capacity less of a consideration.

Architecting for Mixed Object Sizes

As expected, constructing a Ceph cluster for mixed workloads would utilize an architecture somewhere in-between the two extremes outlined for small objects and one designed for larger objects. A moderate amount of CPU and RAM per host would be required, along with multiple SSD's for metadata caching and one or more high speed network (often 25 GbE) interfaces.

Evaluation of Red Hat Ceph Storage

The goal for this project was to configure and test a Red Hat Ceph Storage (RHCS) cluster that could ingest, store and serve 10 Billion small objects with deterministic performance. Red Hat contracted Evaluator Group to test the reference RHCS 4.1 system's ability to support cloud native applications object storage workloads at scale. Red Hat has previously published multiple studies showing the ability for Ceph to scale to large capacities with high performance and multiple nodes up to 1 Billion objects. Moreover, the goal of this project was to push the scale, and potentially the performance by a factor of 10, vs. previous published testing.

Testing was conducted in Evaluator Group labs, using existing equipment along with loaner equipment from contributing sponsors, Intel and Seagate. Seagate provided the high-density JBOD enclosures and 16 TB SAS enterprise nearline disk drives while Intel provided the write optimized NVMe devices used for meta-data caching and memory DIMMS. A detailed configuration is shown in Figure 2 on page 7.

Testing Overview

As an overview of the evaluation, we provide the thesis, or questions to be answered, along with our testing process, our expectations and a brief summary of results. The remainder of the paper along with the Appendix provides significant details on testing results.

Thesis: Does the system performance for common operations remain constant as the capacity and object count increases up to our pre-determined capacity limit of 10 Billion, 64K Objects?

Process: Run multiple test cycles, that performs PUTS only to add to the object count, then another that performs GETS only, followed by a mixed workload test. Measure performance of each successive workload in order to show performance trends as capacity and object count increases.

Expectations: Due to previous experience with similar tests, Red Hat engineering believed that using a relatively large number of HDD and NVMe devices would provide stable performance, until the meta-data cache reached its limit and flushed to the HDDs. The question was at what point would the flushing, or "spilling" of cache begin and what would be the performance impact.

Results: The performance, as measured by objects per second for both PUTS and GETS was better than expected. Additionally, the performance was very stable up until we reached the set capacity of the meta-data / RocksDB cache and cluster spatial capacity.

Test Environment

The configuration and equipment utilized for testing are shown below in Figure 2.

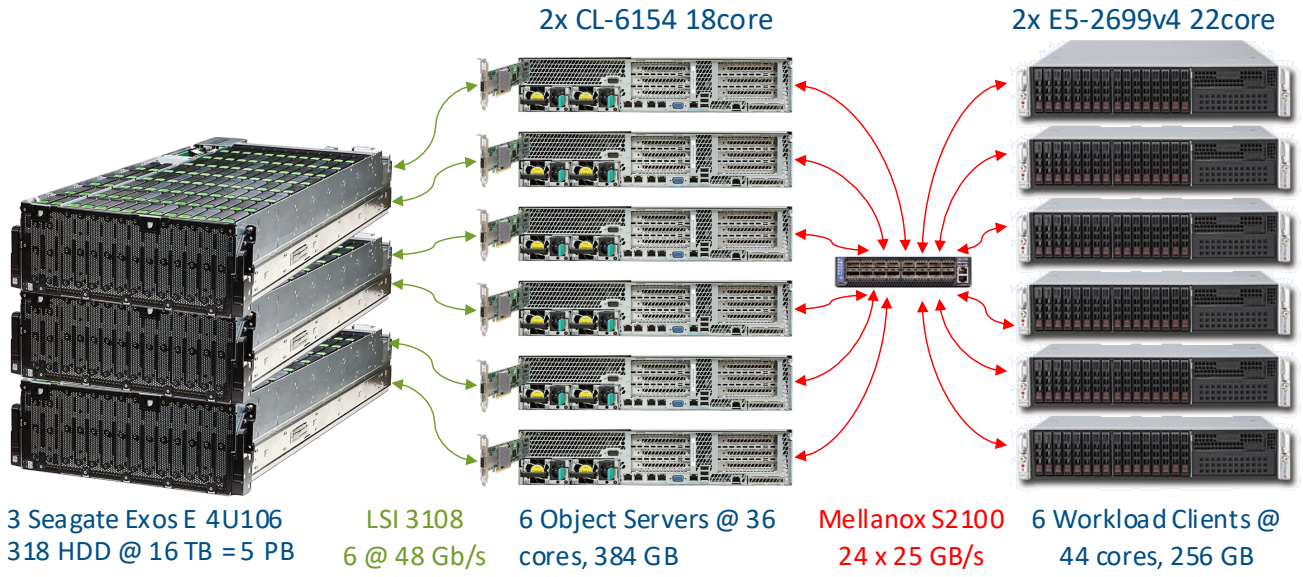

Figure 2: Red Hat Ceph – 10Billion Object Cluster lab

Lab Hardware Overview

Note-1: *Seagate Technology provided the Exos E 4U106 high capacity disk storage arrays and the Exos X16 enterprise SAS disk drives used in this project.*

Note-2: *Intel Corporation provided the P4610 NVMe devices used in this project.*

The test environment consisted of the following elements:

- 3 @ Seagate JBOD Exos E 4U106 enclosures (each shared by two nodes in a split, shared-nothing configuration)
- 318 @ Seagate Exos X16 enterprise nearline SAS 16 TB drives (53 per node)
- 36 @ Intel NVMe 7.6 TB - P4610 (6 per node)
- 6 @ Ceph Storage Nodes
- Intel, 2U – 2 socket server, with 8 NVMe drive support
- 2 x Intel Xeon Scalable 6154 (18 cores / socket), 72 threads
- 384 GB DRAM
- Mellanox ConnectX 5-EN, dual port, 2 x 25 Gb/s
- LSI / Broadcom, 3108 – 8e SAS HBA (single 4x 12 Gb port used)

Architectural Considerations

The configuration utilized for testing was designed to provide good throughput, capacity and high I/O rates simultaneously. We generally followed Red Hat's recommendations, although our design was customized in order to optimize capacity, throughput and object I/O rate (see appendix for customization details). In order to achieve these goals, there were several critical components.

One of the most important elements was the total meta-data capacity. Our test configuration utilized 45 TB of write-intensive NVMe capacity per node allocated for Ceph meta-data. Additionally, a large number of high capacity HDD's per node were utilized, along with sufficient CPU cores, memory and high bandwidth network connections of 50 Gb/s to each Ceph node.

*(**Note**: See Appendix for additional information on Object Storage Node configuration.)*

The Intel CPU provides increased per-core and socket performance relative to prior generations with 48 lanes of PCIe 3.0 per CPU and increased memory bandwidth. The Intel P4610 NVMe SSD's provide consistently low latency under heavy write workloads with a high endurance.

Our Test Configuration had the following design goals:

- Designed for a mix of workloads – small and large objects, GET and PUT operations
- Moderate Cost with dense Capacity
    - Use of Seagate JBOD 4U enclosures to house 4.8 PB in 12 rack units
- High Capacity
    - A total of 318 total HDD's @ 16 TB provided 4.8 PB of raw capacity
- Good performance for small objects
    - Utilized 12 Rados gateways

    - Use of 45 TB of Intel, high performance NVMe drives per node for meta-data
- High Throughput for Objects of 1MB and larger
    - Utilize 50 Gb/s bonded network per node

Red Hat recommends OSD metadata to be at least 4% of the raw capacity of each HDD. Our configuration utilized a slightly greater capacity of approximately 5.5%, due to the size and number of NVMe devices used. The algorithm used by Ceph's RocksDB to dictate when meta-data is pushed (or spilled) out of cache onto a slower tier uses multiple layers, with each being 10X larger than the previous tier. Moreover, in order to significantly delay when data "spills" out of cache would require a 10X increase in amount of cache capacity.

Our test configuration utilized an erasure coded object pool with erasure coding protection to provide fault tolerance while maximizing storage capacity. Use of "EC 4 + 2" was chosen due to its optimal fit with our 6-node configuration while providing full tolerance for the loss of up to 2 nodes, or multiple devices across multiple nodes. Using this scheme resulted in every 64K object split into 4 chunks of 16KB each, and then stored along with two 16KB chunks of parity. Thus, every object stored required 6, I/O operations for every PUT operation.

Workload Scenarios

It is common for the performance of any resource to decrease when exceeding 80% of capacity, although this is a general observation. Ceph recommends not exceeding 80% capacity utilization in order to maintain both good performance and the ability to quickly recover from device failures. Moreover, we chose not to significantly exceed an 80% capacity in recognition of operational best practices during our testing.

Small Object Testing

The primary focus of testing was to understand the systems behavior while running a workload that is comprised of relatively small, 64KB objects which may be used by cloud native or other applications.

For workload generation we utilized the COSbench object storage workload tool. A total of 12 - COSbench clients were used, each using a dedicated Ceph storage target URL. Each "run" consisted of three separate workloads with the following parameters:

- A "write-only" portion of test that generated a total of 76.8 million, 64K objects
- A "read-only" portion that ran for 10 minutes and read as many 64K objects as possible
- A mixed workload that ran for 10 minutes, with Read (70%), Write (20%), List (5%), Delete (5%)

In Figure 3 below, we chart the PUT /sec. and GET / sec. vs. the total number of objects.

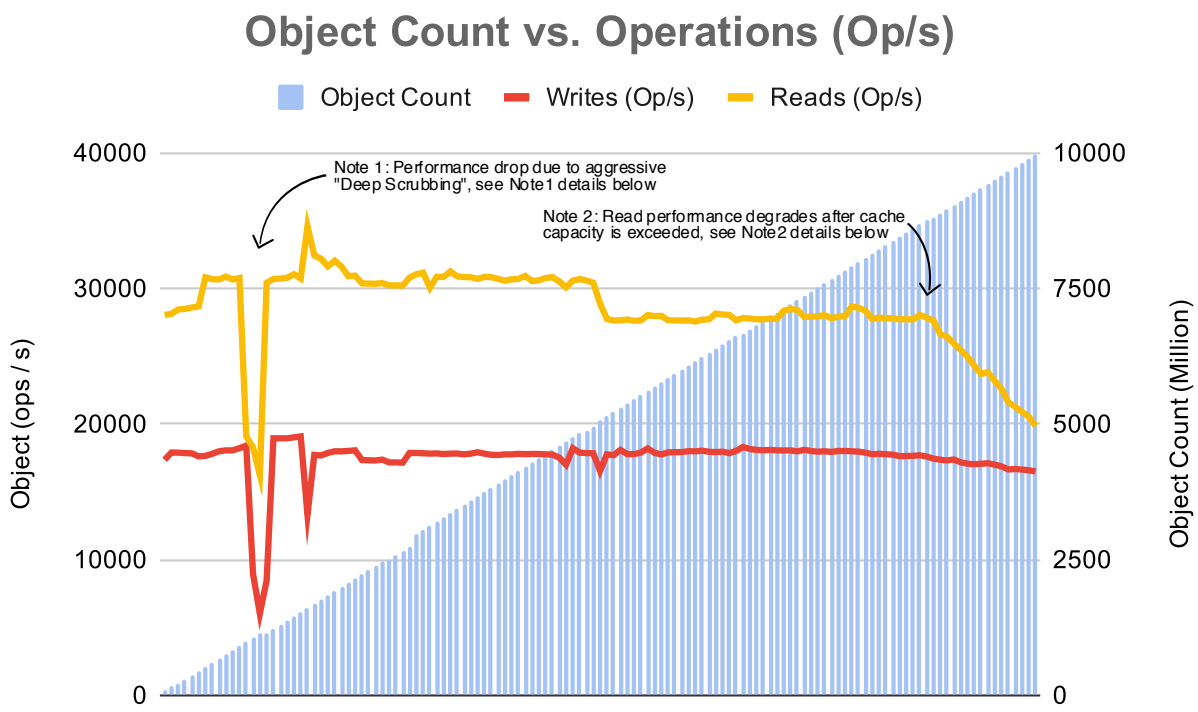

Figure 3: Small Objects, Operations vs. Capacity

Figure 3 - Note 1*: As seen in the performance lines in Figure 3, there was a brief, significant drop in both PUT and GET performance after approximately 500 million objects. This was due to Ceph initiating "deep scrubbing" operations at a high rate, due to the very high "PUT" rates. We reconfigured Ceph to decrease the rate of deep scrubbing, which then returned to normal performance levels. Deep scrubbing is a part of Ceph's architecture to maintain data consistency.*

Figure 3 - Note 2*: As seen on the right side of the performance lines in Figure 3, "GET" operation performance began to degrade after the meta-data cache capacity reached capacity and the cluster usable capacity surpassed 80% of usable capacity. This is explained further below.*

As shown in Figure 3, we plotted two different workload operations, GET (Read) operations and PUT (Write) vs. the object count of the system as it scaled to 10 Billion objects. The PUT operations showed the least variance, with linear performance up until approximately 8.8 Billion objects. GET operations were somewhat linear, but did see a gradual decline, with the drop more pronounced at the same object count.

- Performance for GET operations was consistent up to 8.8 Billion objects

- From 1 to 8.8 Billion objects, the decline was approximately 3.3%
- From 8.8 Billion to 10 Billion objects, performance dropped 29.8%
- Performance for PUT operations was observed to be deterministic
- From 1 to 8.8 Billion objects, performance declined only 0.75%
- From 8.8 Billion to 10 Billion, performance dropped 5.3%

The performance drop for GET operations is expected once the capacity of the Ceph meta-data cache is exceeded. The amount of meta-data is correlated to the number of objects, with each object having a fixed and size dependent amount of meta-data. Once capacity on the NVMe devices was exceeded, some meta-data was pushed or "spilled" to the HDD devices, which results in slower GET operations for those objects. At the 8.8 Billion object level, the system's usable capacity was over 80%, which is typically the recommended operational maximum capacity. However, we continued past this point until we reached our objective of 10 Billion objects, despite the expected performance degradation.

The meta-data generated was approximately 10 KB per 64KB object with the total meta-data generated for 10 Billion, 64K objects approximately 95 TB. The Ceph / RocksDB cache utilization was limited to approximately 275 GB per OSD. With 275 GB * 318 devices, this yields approximately 87 TB of available meta-data cache. Once meta-data capacity on the NVMe cache devices was full, the remaining data was pushed or "spilled" to slower HDD storage devices.

As this test was focused on small, 64K objects, the primary consideration was the object operations per second. However, the throughput or data transfer rates were significant at these object rates, with nearly 2 GB/s sustained throughput for reads, and over 1 GB/s sustained write throughput. This can be seen below in Figure 4.

Large Object Testing

Objects greater than 1 MB are considered large, thus for "Large Object" testing we chose to use 128MB objects, as these are clearly large and are common with some big data workloads such as MapReduce, Spark and Presto. Our testing again utilized workloads with multiple operation types, including PUTS, GETS, LIST and DELETE operations. These operation mixes were run repeatedly as the system capacity increased to over 80% of total capacity in order to ascertain if there were any changes to performance or response times as the system capacity reached operational limits.

The COSbench object storage workload tool was used with the following parameters:

- A total of twelve (12) COSbench clients, each utilizing a dedicated Ceph RGW interface
- A "write-only" PUT portion of test that generated a total of 288 thousand, 128MB objects
    - 288K * 128 MB = 36 TB of new data per workload
- A "read-only" GET portion that ran for 30 minutes and read as many objects as possible

- A "mixed" workload that ran for 30 minutes, and performed 4 operations:
  - GET (70%), PUT (20%), LIST (5%), DELETE (5%)

The performance for GET and PUT operations for 128 MB objects is shown below.

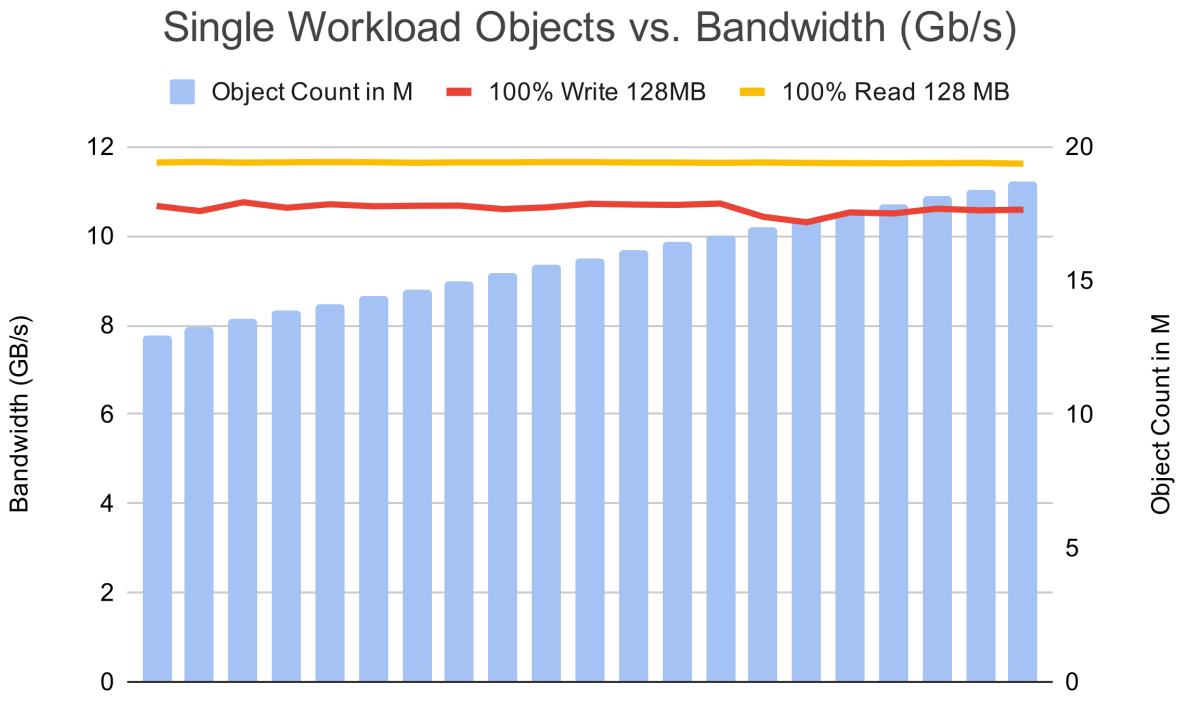

Figure 4: Large Objects Bandwidth vs. Capacity

Above in Figure 4, Ceph showed deterministic performance of more than 10 GB/s for both GET and PUT operations. Again, as seen in the chart the performance was deterministic, with almost no variation as the system filled to 80% of its capacity. Since these large objects generated less meta-data, it did not spill over from HDD to NVMe devices, thus helping to achieve deterministic (i.e. nearly constant) performance as capacity and object counts increased.

As previously discussed, our system design was not optimized for large objects, as there were several known limitations which precluded maximum throughput rates. These included the network bandwidth of 50 Gb/s per node, and the server to storage bandwidth limitation of 48 Gb/s per node over a single SAS connector. If these two components had higher bandwidths, we are confident the system would have achieved significantly higher sustained object throughput rates.

Failure Testing

During testing, the lab experienced a multi-hour power outage over a weekend that exceeded the capacity of the UPS systems. As a result, the entire cluster went down without a graceful shutdown process. The cluster was recovered without any data loss, although data rebuilding did occur for several hours.

The outage occurred at the 5 Billion object level, and although subsequent testing showed nearly identical PUT object rates, we did notice a slight drop in GET object rates as may be seen in Figure 3 on page 10. After restarting testing, we found that the OSD objects were allocated less memory than had been allocated initially. This reduction in cache for OSD devices could explain the modest reduction in performance.

Additionally, several failure tests were performed by Red Hat. Those results are shown in Figures 5 and 6 below. The focus of these tests was to ascertain the cluster performance with a device failure.

***Note**: Evaluator Group did not perform failure testing, nor did we independently validate the results of the tests shown in figures 5 and 6. Red Hat engineers were provided access to the test environment and ran several tests, with those results shown here.*

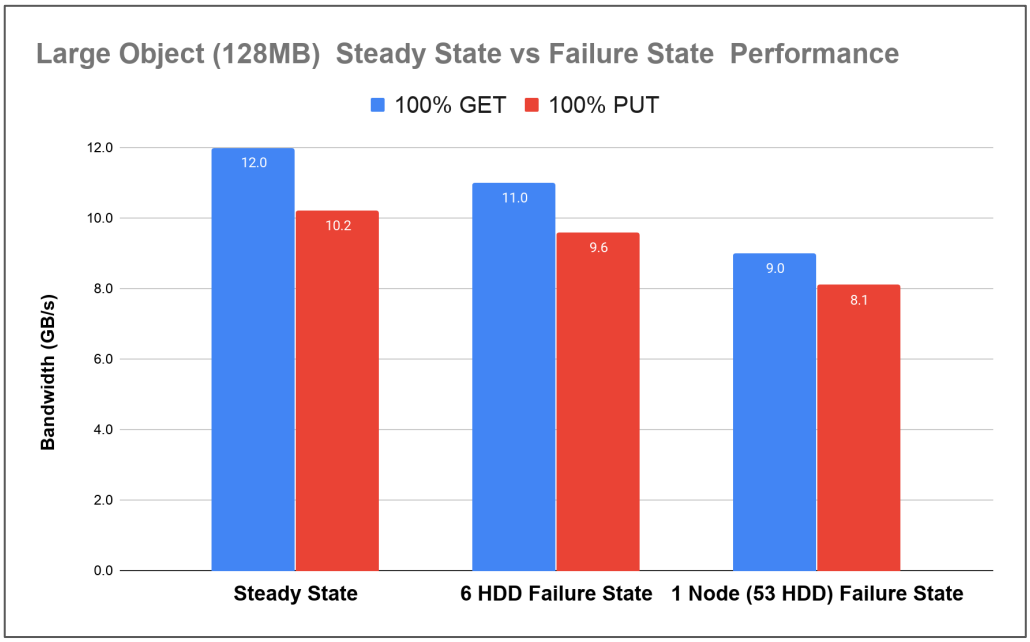

Figure 5: Large Object Performance during OSD failures (Source: Red Hat)

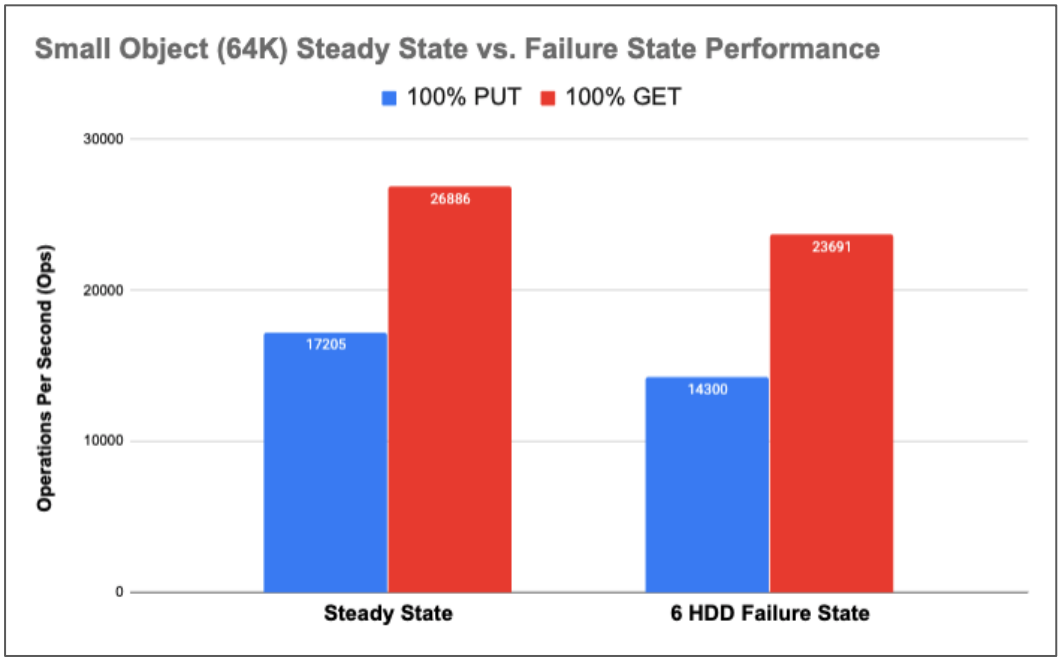

Figure 6: Small Object Performance during OSD failures (Source: Red Hat)

Ceph Testing Summary

Overall, Ceph is highly scalable with many configuration and deployment options. It is possible to create a highly optimized solution for a particular workload, although for general use cases with a mixture of object sizes and access ratios, a configuration that is not overly optimized for any one workload is recommended. The tested configuration provided very good large object bandwidth for both read and write operations while also producing a very high I/O or object rate for smaller objects for a variety of workload mixtures.

The performance of Ceph storage has been evaluated many times, including as graduate level dissertations along with in-depth studies by Red Hat and their team of Ceph architects and developers. Based upon these studies, our evaluation was designed to optimize the architecture in order to provide nearly linear performance as the utilized capacity increased. One of the key architectural considerations for our testing was the number and capacity of the NVMe media utilized for the Ceph meta-data (BlueStore with RocksDB).

Red Hat Ceph architects provided guidance and recommendations on the system configuration, including the critical NVMe components. It was determined that the number and capacity of NVMe devices could store over 80% of the meta-data that would be generated from 10 Billion - 64 KB objects. These calculations proved highly accurate, with performance dropping only after this threshold was reached.

Recommendations

Cloud native workloads are driving the need for storage solutions that can scale both performance and capacity to meet application needs without adding additional complexity and while minimizing costs. Red Hat Ceph Storage is a highly configurable and scalable object storage software offered as a supported product offering. Configurations may scale from 4 nodes up to 100's of nodes, with each system utilizing an appropriate amount of CPU, memory, SSD cache and capacity media to meet a set of application requirements. Ceph's flash-based metadata architecture is able to leverage flash media for sensitive metadata and transactional consistency while utilizing nearline HDD's for capacity.

Our tested configuration was based upon recommended deployments but was modified in order to provide high object I/O rates for small objects while still providing high bandwidth for larger objects. The modifications are documented in the Appendix include the use of multiple Rados gateway devices per node, using port bonding for a higher speed network connection and the use of large capacity NVMe devices by multiple object storage device processes.

In order to be effectively utilized for cloud native applications, an object storage system must have attributes that facilitate use by data scientists, developers, and importantly the system must have features that enable system administrators and operations staff to manage the system.

During our setup and test process, we found that because of the many Ceph configuration and deployment options, it is important to consult with an experienced Ceph architect prior to deployment. When properly configured and deployed, Ceph can provide high performance with relatively little hardware.

For smaller organizations, we would recommend using standard, well proven reference architected solutions. A key consideration for smaller organizations is manageability, due to limited IT staff and budgets. Moreover, the most important factors are ease of use, administration and on-going support. Using proven reference designs should help facilitate stability and simplify management.

For larger organizations looking to utilize Ceph as a key part of their next generation architectures and platforms for cloud native applications, it is recommended to work with Red Hat or other companies with extensive experience in architecting and administering Ceph. Due to the myriad of customization options, it is important to create configurations the meet both application and administrative expectations in order to minimize operational costs.

Appendix

Ceph Configuration Details

The tested configuration did not utilize standard installation scripts or configurations. Typically, Red Hat Ceph 4.1 uses a web-based UI in order to configure settings and parameters, including the number of nodes, along with network interfaces, Ceph monitor nodes and the devices to use for capacity and meta-data devices. NUMA was not configured, and hyperthreading was enabled in the BIOS. Our configuration utilized two Rados gateways per node, although standard Ceph installations utilize a single Rados Gateway process per node.

In summary, standard Ceph 4.1 installations would use the following:

- A single Rados gateway processes per physical node
- The use of a full device for Ceph metadata
- The use of a physical network port for Ceph communication
- Use of separate, physical JBODs for each Ceph storage node
- Ceph's deep scrubbing priority left at default (our configuration used a lower priority)

The tested configuration used the following settings:

- Two Rados gateway processes per physical node, with the web service running on different ports, rather than the default one Rados gateway process
- The use of partitions for the Ceph, Bluestore metadata storage on NVMe devices. By using 6 NVMe devices with 9 partitions, this provided the required 53 meta-data partitions required 53 OSD device processes
- The use of a dedicated NVMe partition for each OSD process, resulting in 9 OSD processes sharing a single P4610 NVMe device
- The use of network port bonding, with 2 x 25 Gb/s interfaces bonded together for one 50 Gb/s interface, using Red Hat OS bonding in LACP mode, along with LACP configuration on the Mellanox SN2100 switch ports
- The external Seagate Exos E 4U106 JBOD was logically "split" providing two separate 53 device units. With one physical Ceph node using each of the split JBODs. Thus, the three physical JBODs appeared as six logical units
- Ceph's deep scrubbing was set to a lower priority and scrub rate than the default during the testing, which appears as the "Note 1" on the performance graphs

Object Storage Server Configuration

To achieve the highest utilization possible of the NVMe drives, high core-count CPUs are used to increase the level of parallelism and performance within the storage node. Ceph OSDs contain many active threads, and more cores allow more processes to run, increasing transactional throughput.

The CPU microarchitecture of the Intel Xeon scalable processor servers utilized as the Object storage servers provided increased per-core and socket-level performance and has 48 lanes of PCIe 3.0 per CPU compared to previous system generations.

The P4610 meta data devices on the object storage systems provide fast response times under heavy load with an endurance rating of 3.157 DWPD. The P4610 devices have SMART attributes E2, E3, and E4 for off-line endurance evaluations as well as E9, E8 as real-time/production life/ endurance indicators. SMART attributes are used for monitoring SSD health status, such as endurance indicator, error log, host read/write, NAND read/write. NVMe has low latency due to the elimination of several intermediate I/O functions, along with bandwidth PCIe bus.

Additional Test Results

Shown in Figure 7 is the bandwidth for the small object workloads.

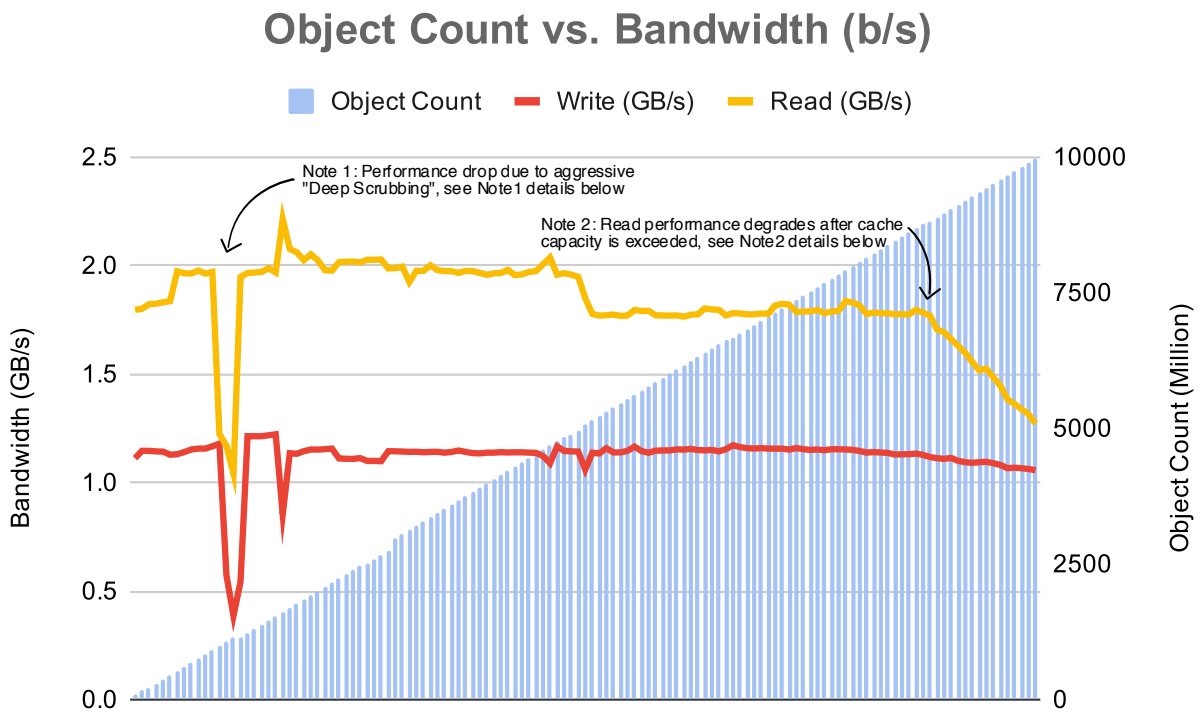

Figure 7: Small Objects (64KB) Bandwidth vs. Capacity

The LIST and DELETE operations along with GET and PUT operations were run in a mixed workload to ascertain any performance changes over time. The relative performance is not particularly interesting, in that the workloads were run with specific ratio's, as previously noted. Thus, PUT, LIST and DELETE operations were expected to have few operations, as defined by the workload. Only their change over time and spatial capacity is relevant.

A mixed workload was run that included GET, PUT, LIST and DELETE operations. The ratio was (70% GET, 20% PUT, 5% List and 5% Delete). Since the ratios were fixed, the performance changed proportionally for the workloads. This is shown below in Figure 8.

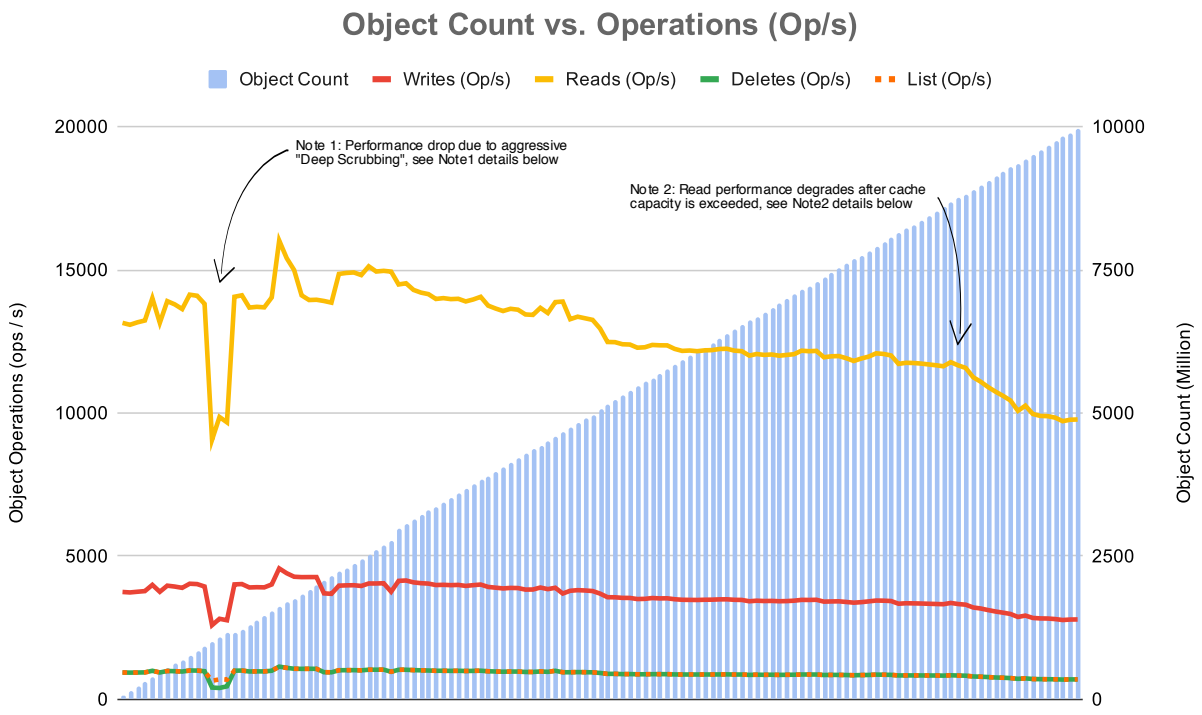

Figure 8: Small Objects (64KB) Mixed Workload Operations / sec

As seen above, performance was nearly constant for most operations, with some performance variations as noted in the graphic. In particular, the following observations are noted:

- The performance for the mixed workload shows nearly constant performance for each PUT, LIST and DELETE operations as capacity and object counts increased.
- There was a gradual performance drop for GET (read) operations as capacity increased while other operations appeared less effected by the growing capacity.

About Evaluator Group

*Evaluator Group Inc. is dedicated to helping **IT professionals** and vendors create and implement strategies that make the most of the 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 the positions of such vendors on the issues addressed in this document.