Dell EMC Ready Architecture for Red Hat Ceph Storage 3.2

Object Storage Architecture Guide

Dell EMC Service Provider Solutions

Contents

List of Figures

List of Tables

Trademarks

Copyright © 2014-2019 Dell Inc. or its subsidiaries. All rights reserved.

Microsoft® and Windows® are registered trademarks of Microsoft Corporation in the United States and/or other countries.

Red Hat®, Red Hat Enterprise Linux®, and Ceph are trademarks or registered trademarks of Red Hat, Inc., registered in the U.S. and other countries. Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Intel® and Xeon® are registered trademarks of Intel Corporation. Oracle® and Java® are registered trademarks of Oracle Corporation and/or its affiliates.

Cumulus®, Cumulus Networks®, Cumulus Linux®, is a registered trademark of Cumulus, registered in the U.S. and other countries.

Amazon® and S3® are registered trademarks of Amazon.com, Inc.

DISCLAIMER: The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation or the OpenStack community.

Notes, Cautions, and Warnings

- ✎ A **Note** indicates important information that helps you make better use of your system.
- ⚠ A **Caution** indicates potential damage to hardware or loss of data if instructions are not followed.
- ⚠ A **Warning** indicates a potential for property damage, personal injury, or death.

This document is for informational purposes only and may contain typographical errors and technical inaccuracies. The content is provided as is, without express or implied warranties of any kind.

Chapter 1

Introduction

Topics:

Dell EMC has several different Ready Architectures for Red Hat Ceph Storage 3.2 that are designed and optimized to fulfill different objectives. There are architectures for:

- Cost-optimized and balanced block storage with a blend of SSD and NVMe storage to address both cost and performance considerations
- Performance-optimized block storage with all NVMe storage
- Performance- and capacity-optimized object storage, with a blend of HDD and Intel® Optane® storage to provide high-capacity, excellent performance, and cost-effective storage options

This document covers the **Dell EMC Ready Architecture for Red Hat Ceph Storage 3.2 for Performance and Capacity Optimized Object Storage**.

This chapter provides an overview of Red Hat Ceph Storage software and of the key Dell EMC hardware components used in the solution, including the Dell PowerEdge R740xd storage server and the Dell EMC PowerSwitch S5248 switch. Subsequent chapters provider further details on Red Hat Ceph Storage, the architectural design and components, and the test setup and validation methodologies that we have used to validate the design and performance.

Introduction

Unstructured data has demanding storage requirements across the access, management, maintenance, and particularly the scalability dimensions. To address these requirements, Red Hat Ceph Storage provides native object-based data storage and enables support for object, block, and file storage. Some of the properties are shown in the diagram below.

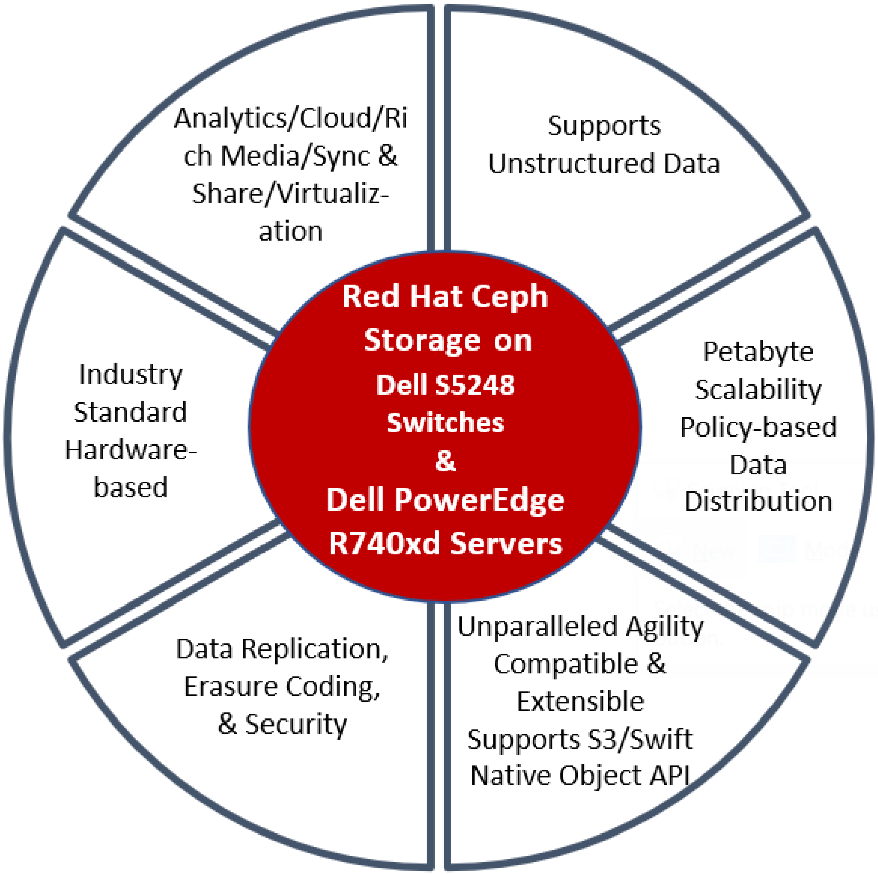

Analytics/Cloud/Rich Media/Sync & Share/Virtualiz-ation
Supports Unstructured Data
Industry Standard Hardware-based
Red Hat Ceph Storage on Dell S5248 Switches & Dell PowerEdge R740xd Servers
Petabyte Scalability Policy-based Data Distribution
Data Replication, Erasure Coding, & Security
Unparalleled Agility Compatible & Extensible Supports S3/Swift Native Object API

Figure 1: Key takeaways of deploying Red Hat Ceph Storage on Dell EMC PowerEdge R740xd servers

The Red Hat Ceph Storage environment makes use of industry standard servers that form Ceph nodes for scalability, fault-tolerance, and performance. Data protection methods play a vital role in deciding the total cost of ownership (TCO) of a solution. Ceph allows the user to set different data protection methods on different storage pools.

Dell PowerEdge R740xd

The PowerEdge R740xd delivers a perfect balance between storage scalability and performance. The 2U two-socket platform is ideal for Software-defined storage (SDS), service providers or as Virtual desktop infrastructure (VDI).

The scalable system architecture behind the R740xd with up to 24 NVMe drives creates the ideal balance between scalability and performance. The R740xd versatility is highlighted with the ability to mix any drive type to create the optimum configuration of NVMe, SSD and HDD for either performance, capacity or both.

The Dell PowerEdge R740xd offers advantages that include the ability to drive peak performance by:

- Maximizing storage performance with up to 24 NVMe drives and ensures application performance scales to meet demands.
- Freeing up storage space using internal M.2 SSDs optimized for boot.
- Accelerates workloads with up to 3 double-width 300W GPUs, up to 6 single-width 150W GPUs or up to 4 FPGAs.

Dell EMC PowerSwitch S5248F-ON

The S5248 comprises Dell EMC's latest disaggregated hardware and software data center networking solutions, providing state-of-the-art, high-density 25/100GbE ports and a broad range of functionality to meet the growing demands of today's data center environment. It is an ideal choice for organizations looking to enter the software-defined data center era with a choice of networking technologies designed to maximize flexibility.

For applications such as software-defined storage (SDS) requiring the highest bandwidth, the multi-functional 25/100GbE switch is very well suited. This switch can provide high-density Top of Rack (ToR) server aggregation in high-performance data center environments at the desired fabric speed. Some of the features are:

- 1U high-density ToR switch with up to 48 ports of 25GbE, 4 100GbE and 2 200GbE ports
- Multi-rate 100GbE ports support 100/50/40/25/10GbE
- Line-rate performance via non-blocking switch fabric up to 2.0Tbps
- L2 multipath support via Virtual Link Trunking (VLT) and Routed VLT

Chapter 2

Overview of Red Hat Ceph Object Storage

Topics:

This chapter introduces the Red Hat software defined storage (SDS) solution Red Hat Ceph Storage (RHCS). It explains the Ceph terminology like pools, placement groups and CRUSH rulesets. Furthermore, it provides details on how to select various components of the solution, including storage access methods and storage protection methods. Finally, it also introduces the new storage backend BlueStore and highlights its features.

Overview of Red Hat Ceph Storage

A Ceph storage cluster is built from a number of Ceph nodes for scalability, fault-tolerance, and performance. Each node is based on industry-standard hardware and uses intelligent Ceph daemons that communicate with each other to:

- Store and retrieve data
- Replicate data
- Monitor and report on cluster health
- Redistribute data dynamically (remap and backfill)
- Ensure data integrity (scrubbing)
- Detect and recover from faults and failures

A few advantages of Red Hat Ceph Storage are:

- Recognized industry leadership in open source software support services and online support
- Only stable, production-ready code, vs. a mix of interim, experimental code
- Consistent quality; packaging available through Red Hat Satellite
- Well-defined, infrequent, hardened, curated, committed 3-year lifespan with strict policies
- Timely, tested patches with clearly-defined, documented, and supported migration path
- Backed by Red Hat Product Security
- Red Hat Certification and Quality Assurance Programs
- Red Hat Knowledgebase (articles, tech briefs, videos, documentation), and Automated Services

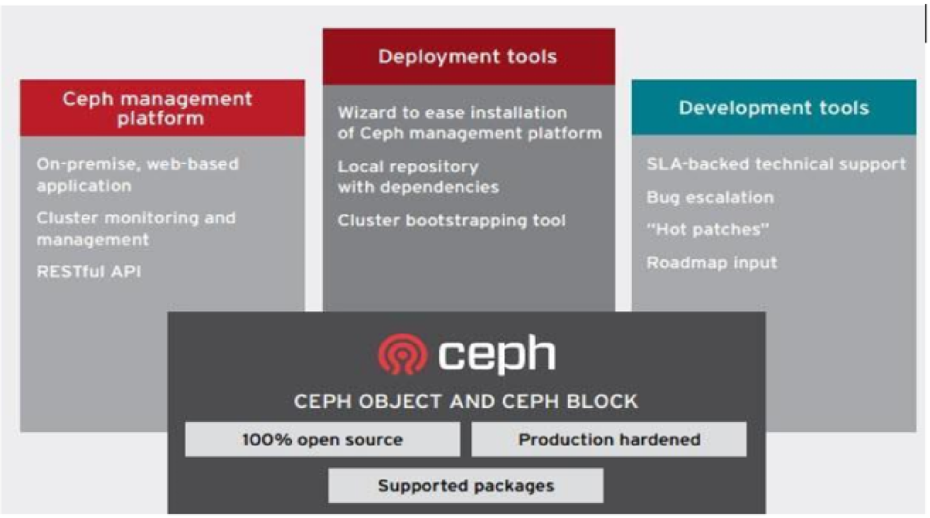
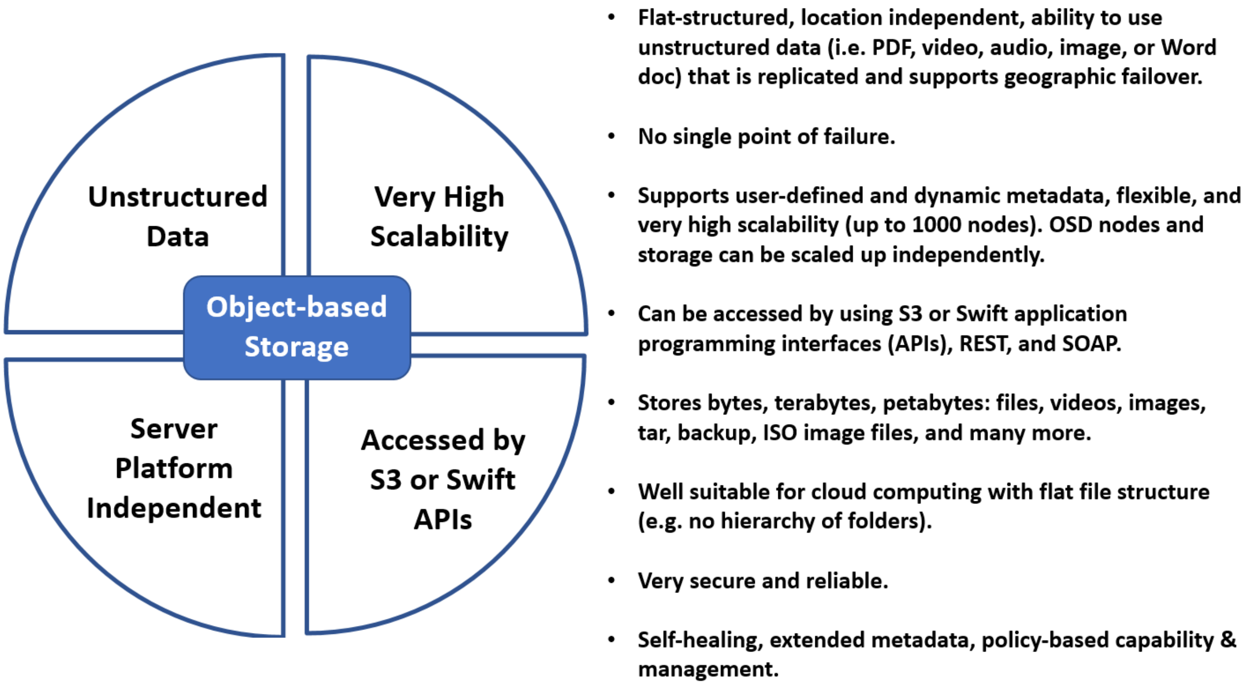

Ceph management platform
On-premise, web-based application
Cluster monitoring and management
RESTful API
Deployment tools
Wizard to ease installation of Ceph management platform
Local repository with dependencies
Cluster bootstrapping tool
Development tools
SLA-backed technical support
Bug escalation
"Hot patches"
Roadmap input
ceph
CEPH OBJECT AND CEPH BLOCK
100% open source
Production hardened
Supported packages

Figure 2: Red Hat Ceph Storage

Red Hat Ceph Storage significantly lowers the cost of storing enterprise data and helps organizations manage exponential data growth. The software is a robust, petabyte-scale storage platform for those deploying public or private clouds. As a modern storage system for cloud deployments, Red Hat Ceph Storage offers mature interfaces for enterprise block and object storage, making it well suited for active archive, rich media, and cloud infrastructure workloads like OpenStack. Delivered in a unified self-healing and self-managing platform with no single point of failure, Red Hat Ceph Storage handles data management so businesses can focus on improving application availability. Some of the properties include:

- Scaling to petabytes
- No single point of failure in the cluster
- Lower capital expenses (CapEx) by running on industry-standard server hardware
- Lower operational expenses (OpEx) by self-managing and self-healing



- Flat-structured, location independent, ability to use unstructured data (i.e. PDF, video, audio, image, or Word doc) that is replicated and supports geographic failover.
- No single point of failure.
- Supports user-defined and dynamic metadata, flexible, and very high scalability (up to 1000 nodes). OSD nodes and storage can be scaled up independently.
- Can be accessed by using S3 or Swift application programming interfaces (APIs), REST, and SOAP.
- Stores bytes, terabytes, petabytes: files, videos, images, tar, backup, ISO image files, and many more.
- Well suitable for cloud computing with flat file structure (e.g. no hierarchy of folders).
- Very secure and reliable.
- Self-healing, extended metadata, policy-based capability & management.

Figure 3: Ceph object-based storage

Table 1: Ceph cluster design considerations on page 15 below provides a matrix of different Ceph cluster design factors, optimized by workload category. Please see *https://access.redhat.com/documentation/en-us/red_hat_ceph_storage/3/html/configuration_guide/* for more information.

Table 1: Ceph cluster design considerations

Optimization criteria	Potential attributes	Example uses
Capacity-optimized	• Lowest cost per TB • Lowest BTU per TB • Lowest watt per TB • Meets minimum fault domain recommendation (single server is less than or equal to 15% of the cluster)	• Typically object storage • Erasure coding common for maximizing usable capacity • Object archive • Video, audio, and image object archive repositories
Throughput-optimized	• Lowest cost per given unit of throughput • Highest throughput • Highest throughput per Watt • Meets minimum fault domain recommendation (single server is less than or equal to 10% of the cluster)	• Block or object storage • 3x replication • Active performance storage for video, audio, and images • Streaming media

Introduction to Ceph storage pools

For a Ceph client, the storage cluster is very simple. When a Ceph client reads or writes data it connects to a logical storage pool in the Ceph cluster.

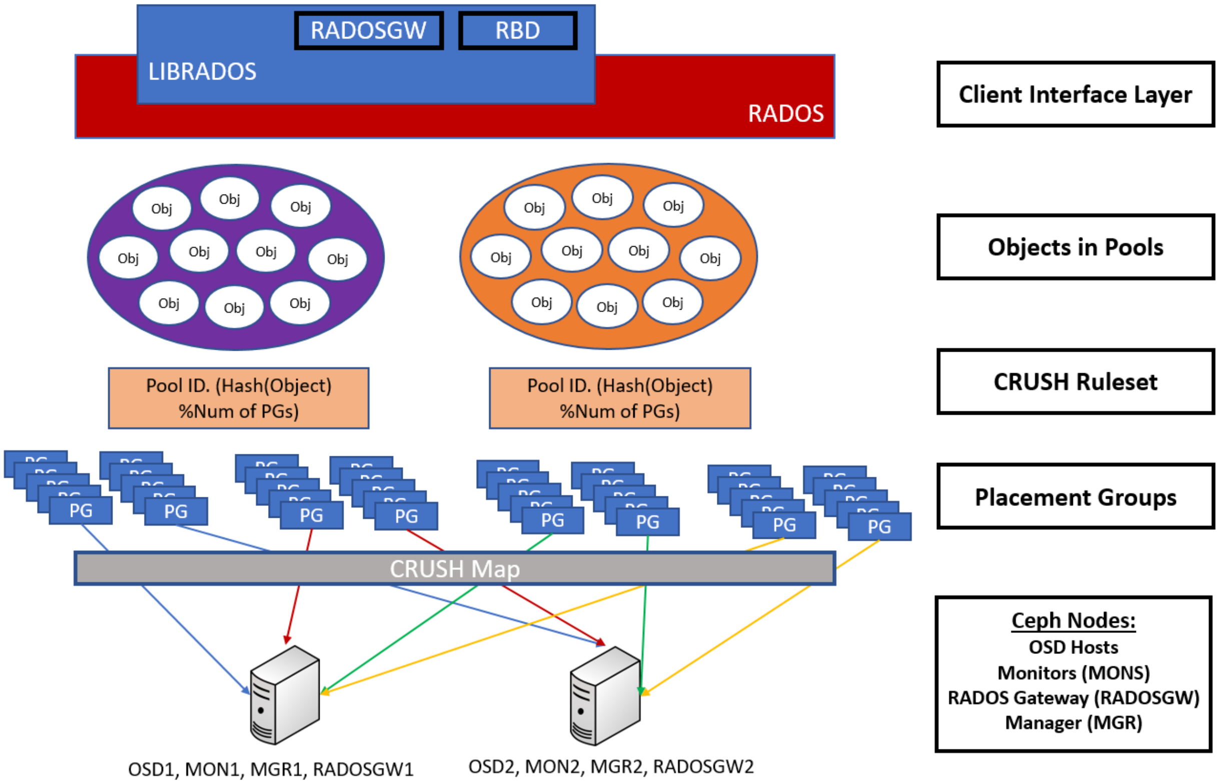


Figure 4: Ceph storage pools

Pools

A Ceph storage cluster stores data objects in logical dynamic partitions called pools. Pools can be created for particular data types, such as for block devices, object gateways, or simply to separate user groups. The Ceph pool configuration dictates the number of object replicas and the number of placement groups (PGs) in the pool. Ceph storage pools can be either replicated or erasure-coded, as appropriate for the application and cost model. Also, pools can “take root” at any position in the CRUSH hierarchy (see below), allowing placement on groups of servers with differing performance characteristics, encouraging storage to be optimized for different workloads.

Placement Groups

Ceph maps objects to Placement Groups (PGs). PGs are shards or fragments of a logical object pool that are composed of a group of Ceph OSD daemons that are in a peering relationship. Placement groups provide a way to creating replication or erasure coding groups of coarser granularity than on a per-object basis. A larger number of placement groups (for example, 200/OSD or more) leads to better balancing.

CRUSH Rulesets

CRUSH is an algorithm that provides controlled, scalable, and decentralized placement of replicated or erasure-coded data within Ceph and determines how to store and retrieve data by computing data storage locations. CRUSH empowers Ceph clients to communicate with OSDs directly, rather than through a centralized server or broker. By determining a method of storing and retrieving data by algorithm, Ceph avoids a single point of failure, a performance bottleneck, and a physical limit to scalability.

Ceph Monitors (MONs)

Before Ceph clients can read or write data, they must contact a Ceph MON to obtain the current cluster map. A Ceph storage cluster can operate with a single monitor, but this introduces a single point of failure. For added reliability and fault tolerance, Ceph supports an odd number of monitors in a quorum (typically three for small to mid-sized clusters, or five for large clusters). Consensus among various monitor instances ensures consistent knowledge about the state of the cluster.

Ceph OSD Daemons

In a Ceph cluster, Ceph OSD daemons store data and handle data replication, recovery, backfilling, and rebalancing. They also provide some cluster state information to Ceph monitors by checking other Ceph OSD daemons with a heartbeat mechanism. A Ceph storage cluster configured to keep three replicas of every object requires a minimum of three Ceph OSD daemons, two of which need to be operational to successfully process write requests.

Client Interface Layer

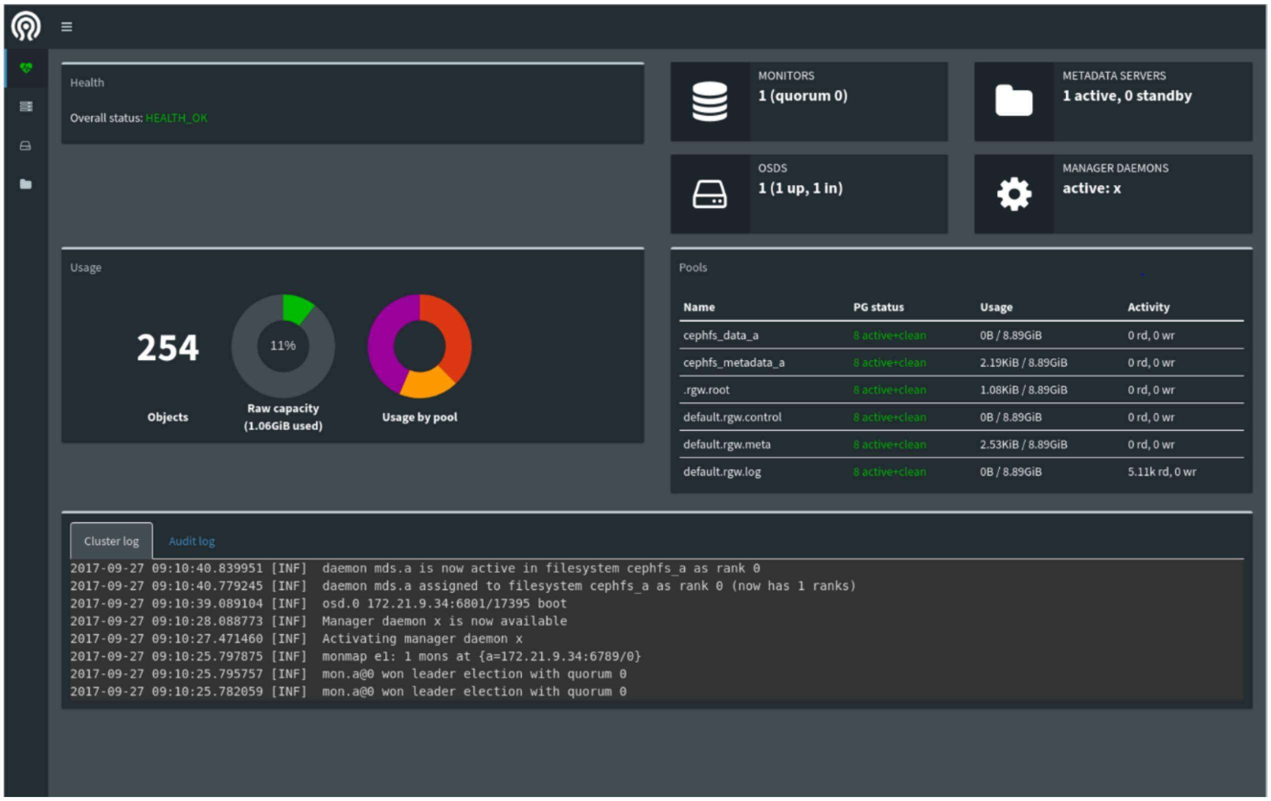
Ceph supports a range of storage methods. Ceph client architecture provides the interface for writing and reading data in a Ceph storage cluster. LIBRADOS provides direct access to RADOS with libraries for most programming languages. For both physical and virtual systems, RBD offers a Ceph block storage device that maps like a physical storage drive. An object storage gateway service, RADOS gateway (RADOSGW), is an object storage gateway service that provides OpenStack® Swift compatible RESTful and S3-compatible interfaces.

Ceph Dashboard

Ceph Manager has the ability to record many Ceph metrics including the throughput, latency, disk usage, cluster health, and others. Ceph Dashboard is a WebUI which can be used to monitor a Ceph cluster. It is powered by the Ceph Manager and provides a detailed visualization of Ceph metrics and cluster status. It's very easy to set up and is available out of the box when Ceph is deployed. It is ideal for monitoring a Ceph cluster with minimum setup effort.

The dashboard currently provides the following features to monitor various aspects of a Ceph cluster:

- Username/password protection
- SSL/TLS support
- Overall cluster health
- Cluster logs
- Hosts
- Performance counters
- Monitors
- Configuration Reference
- Pools
- OSDs
- iSCSI, an Internet Protocol (IP) based storage networking standard for linking data storage facilities.
- RADOS Block Devices (RBD) and RBD mirroring
- Ceph Filesystem (CephFS)
- Object Gateway



Health

Overall status: HEALTH_OK

MONITORS
1 (quorum 0)

METADATA SERVERS
1 active, 0 standby

OSDS
1 (1 up, 1 in)

MANAGER DAEMONS
active: x

Pools

Name	PG status	Usage	Activity
cephfs_data_a	8 active+clean	0B / 8.89GiB	0 rd, 0 wr
cephfs_metadata_a	8 active+clean	2.19KiB / 8.89GiB	0 rd, 0 wr
.rgw.root	8 active+clean	1.08KiB / 8.89GiB	0 rd, 0 wr
default.rgw.control	8 active+clean	0B / 8.89GiB	0 rd, 0 wr
default.rgw.meta	8 active+clean	2.53KiB / 8.89GiB	0 rd, 0 wr
default.rgw.log	8 active+clean	0B / 8.89GiB	5.11k rd, 0 wr

```
2017-09-27 09:10:40.839951 [INF]  daemon mds.a is now active in filesystem cephfs_a as rank 0
2017-09-27 09:10:40.779245 [INF]  daemon mds.a assigned to filesystem cephfs_a as rank 0 (now has 1 ranks)
2017-09-27 09:10:39.089104 [INF]  osd.0 172.21.9.34:6801/17395 boot
2017-09-27 09:10:28.088773 [INF]  Manager daemon x is now available
2017-09-27 09:10:27.471460 [INF]  Activating manager daemon x
2017-09-27 09:10:25.797875 [INF]  monmap e1: 1 mons at {a=172.21.9.34:6789/0}
2017-09-27 09:10:25.795757 [INF]  mon.a@0 won leader election with quorum 0
2017-09-27 09:10:25.782059 [INF]  mon.a@0 won leader election with quorum 0
```

Figure 5: Ceph dashboard

Selecting storage access method

Choosing a storage access method is an important design consideration. As discussed, all data in Ceph is stored in pools, regardless of data type. The data itself is stored in the form of objects using the Reliable Autonomic Distributed Object Store (RADOS) layer which:

- Avoids a single point of failure
- Provides data consistency and reliability
- Enables data replication and migration
- Offers automatic fault-detection and recovery

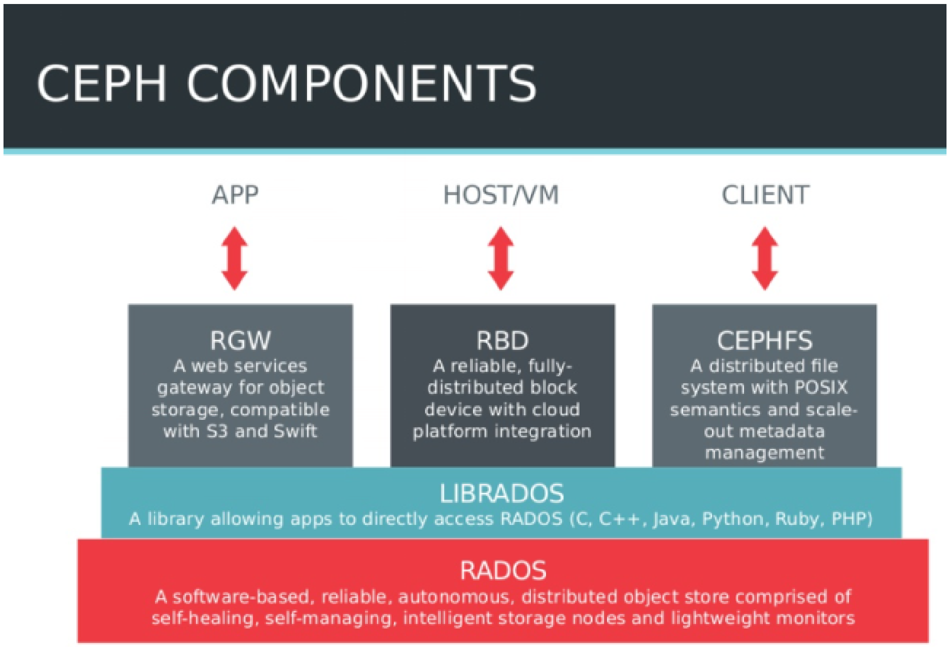

CEPH COMPONENTS
APP
HOST/VM
CLIENT
RGW
A web services gateway for object storage, compatible with S3 and Swift
RBD
A reliable, fully-distributed block device with cloud platform integration
CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management
LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)
RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

Figure 6: RADOS layer in the Ceph architecture

Writing and reading data in a Ceph storage cluster is accomplished using the Ceph client architecture. Ceph clients differ from competitive offerings in how they present data storage interfaces. A range of access methods are supported, including:

- **RADOSGW** Object storage gateway service with S3 compatible and OpenStack Swift compatible RESTful interfaces
- **LIBRADOS** Provides direct access to RADOS with libraries for most programming languages, including C, C++, Java, Python, Ruby, and PHP
- **RBD** Offers a Ceph block storage device that mounts like a physical storage drive for use by both physical and virtual systems (with a Linux® kernel driver, KVM/QEMU storage backend, or userspace libraries)
- **CephFS** The Ceph Filesystem (CephFS) is a POSIX-compliant filesystem that uses LIBRADOS to store data in the Ceph cluster, which is the same backend used by RADOSGW and RBD.

The storage access method and data protection method (discussed later) are interrelated. For example, Ceph block storage is currently only supported on replicated pools, while Ceph object storage is allowed on either erasure-coded or replicated pools.

Selecting storage protection method

As a design decision, choosing the data protection method can affect the solution's total cost of ownership (TCO) more than any other factor. This is because the chosen data protection method strongly affects the amount of raw storage capacity that must be purchased to yield the desired amount of usable storage capacity. Applications have diverse needs for performance and availability. As a result, Ceph provides data protection at the storage pool level.

Replicated Storage Pools

Replication makes full copies of stored objects, and is ideal for quick recovery. In a replicated storage pool, Ceph configuration defaults to a replication factor of three, involving a primary OSD and two secondary OSDs. If two of the three OSDs in a placement group become unavailable, data may be read, but write operations are suspended until at least two OSDs are operational.

Erasure-coded Storage Pools

Erasure coding provides a single copy of data plus parity, and it is useful for archive storage and cost-effective durability and availability. With erasure coding, storage pool objects are divided into chunks using the n=k+m notation, where k is the number of data chunks that are created, m is the number of coding chunks that will be created to provide data protection, and n is the total number of chunks placed by CRUSH after the erasure coding process. So for instance, n disks are needed to store k disks worth of data with data protection and fault tolerance of m disks.

While Ceph block storage is typically configured with 3x replicated pools, Ceph object storage is frequently configured to use erasure-coded pools. Depending on the performance needs and read/write mix of an object storage workload, an erasure-coded pool can provide an extremely cost effective solution while still meeting performance requirements.

See the Ceph documentation at http://docs.ceph.com/docs/master/architecture/ for more information.

BlueStore

BlueStore is a new backend for the OSD daemons that was introduced in the 'Luminous' release of Ceph. Compared to the traditionally used FileStore backend, BlueStore allows for storing objects directly on raw block devices, bypassing the file system layer. This new backend improves the performance of the cluster by removing the double-write penalty inherent in FileStore.

Table 2: BlueStore/FileStore comparison

OSD backend	Data storage	Internal metadata	Journaling
FileStore	Files within XFS file system	XFS metadata	Journal
BlueStore	Raw volume (no file system)	RocksDB	Write-Ahead Log (WAL)

BlueStore provides the following features and benefits:

- Direct management of storage devices
- Metadata management with RocksDB
- Full data and metadata checksumming
- Inline compression
- Efficient copy-on-write
- No large double-writes
- Multi-device support

Selecting a hardware configuration

As a design decision, choosing the appropriate hardware configuration can have significant effects on the solution’s performance and cost. Applications have diverse requirements for performance and different available hardware budgets. To meet these varying requirements, we provide an architecture that is optimized for capacity and performance. This architecture is presented in the next chapter.

Chapter 3

Architecture components

Topics:

This chapter introduces the starter 4-node, 50GbE cluster with containerized Ceph daemons and discusses the rationale for the design. The choices of hardware and software components, along with deployment topology are presented with an explanation of how they support the architectural objectives.

Note: Please contact your Dell EMC representative for sizing guidance beyond this starter kit.

Architecture overview

With an ever-increasing demand of performance- and capacity-optimized storage with a parallel need for scalability, we designed an architecture that delivers high capacity while ensuring good performance. The 3.5" HDDs provide high capacity, while the use of Intel® Optane® high speed storage devices assist in providing very good performance. Moreover, the reduced number of nodes due to colocation makes an immense reduction in CapEx and OpEx making it an ideal choice for small to medium sized data centers. Ceph's scalability and our network design allows for cluster expansion while still providing high capacity, hardware utilization, and performance.

The architecture presented in this chapter was designed to meet the following objectives:

- High capacity
- Very good performance
- Cost effective
- High availability
- Leverage Ceph 3.2 improvements
- Easy to administer

Traditionally, a Ceph cluster consists of any number of storage nodes (for OSD daemons), and three additional nodes to host MON daemons. While the MON daemons are critical for functionality, they have a very small resource footprint. Red Hat Ceph Storage (RHCS) 3 introduced the ability to run Ceph daemons as containerized services. With this, the colocation of MON and OSD daemons on the same server is a supported configuration. This eliminates the need for additional dedicated MON nodes and provides us with a significant reduction in cost.

Since the architecture was designed for general-purpose object storage with high availability, the components were carefully selected and designed to provide an architecture that is performance- and capacity-optimized, yet affordable and flexible. This, along with the fact that RHCS 3.2 has a much-improved storage backend BlueStore, among other enhancements, allows us to get the most out of the hardware. Finally, the design was also optimized to ensure sustained performance with various workloads. This makes it a perfect fit for a wide array of use cases in production environments that require scalability while keeping cost increase to a minimum.

R740xd Storage Node

We chose the Dell EMC PowerEdge R740xd as it provides the best balance of PCIe slot availability, capacity for internal drives, and performance for Ceph storage nodes. The R740xd server has chassis options to support 3.5" drives and another to support 2.5" drives. The 3.5" drive chassis was selected for this architecture because the 3.5" drives provide the highest capacity while maintaining cost effectiveness.

Storage devices

We chose **10TB (see caution) SATA HDDs** for data storage since it provides a nice balance of high capacity and relatively low cost. **Intel® Optane® (see note) P4800X** was chosen to provide high-speed storage for BlueStore WAL and RocksDB metadata.

Table 3: R740xd storage devices

Drive usage	Drive description	Quantity	Drive capacity
Ceph data (OSD)	7.2K RPM SATA 3.5" HDD	12	10 TB* (see caution)

Drive usage	Drive description	Quantity	Drive capacity
Ceph metadata (BlueStore WAL/RocksDB; bucket indexes)	Intel® Optane® (see note) P4800X	2	750 GB

Note: Our performance testing was conducted with P4600 NVMe devices because the P4800X was not orderable at the time the servers were acquired. Please use P4800X instead of P4600.

CAUTION: Please consult with Red Hat to discuss support for use of drives with capacity above 8TB.

Disk controller

Although RAID arrays (multiple devices within a single array) are not appropriate for Ceph, a RAID controller can still be used if it allows "pass-through" mode or is configured with each disk in its own RAID-0 configuration. Some may choose to use a RAID controller over JBOD controller for the purpose of the RAID controller's on-board cache. Internal testing has shown that the RAID controller's cache can be helpful in some scenarios at lower loads. These scenarios are highly dependent on workload characteristics. Additionally, we have observed that there is no benefit of the cache in the presence of heavy workloads. The downside of the controller's cache is that it can give less predictable ("lumpy") performance. For these reasons, we use the **HBA330 JBOD controller** in this architecture and generally recommend it over a RAID controller for Ceph workloads.

Note: We generally recommend the HBA330 JBOD controller instead of a RAID controller for Ceph workloads.

Another consideration is the system disks used for the operating system. The system disks are commonly set up in a RAID-1 array for fault tolerance. Dell's Boot Optimized Storage Solution (BOSS) provides this functionality on a single PCIe card with embedded M.2 units established in a RAID-1 array. We use this BOSS card in the storage nodes of our architecture to provide fault tolerant operating system storage.

CPU and memory sizing

As noted above, the architecture is designed with 12 OSDs per node. Current best practices suggest 16 GB of base memory for the OS, with a minimum of 2 GB per OSD. Additionally, it is suggested that a minimum of 1 GB be allocated for each additional Ceph daemon.

Table 4: Sizing memory requirements

Component	Min. RAM per instance (GB)	Recommended RAM per instance (GB)	Instances	Total Min. RAM (GB)	Total Recommended RAM (GB)
Operating system	16	16	1	16	16
Ceph OSD	2	8	12	24	96
Ceph MON	1	1	1	1	1
Ceph MGR	1	1	1	1	1
Ceph RGW	1	1	1	1	1
Total	-	-	-	43	115

The table above illustrates that 43 GB is the minimum memory requirement, with 115 GB as the recommended memory configuration for each storage node. The best performance for memory access in Dell PowerEdge servers is obtained by having all slots in the first memory bank of each CPU populated equally. The R740xd contains a total of 24 memory (DIMM) slots split equally among 2 CPU sockets. The CPU provides six memory channels and the first bank of six slots plug directly into the six CPU memory channels. Since server memory is typically installed in increments of 16 or 32 GB, high performance memory access is achieved by populating each CPU's first memory bank with six 16 GB DIMMs for a total of 192 GB.

Note: Populating all six DIMM slots of each CPU's first memory bank (12 total for both CPUs) provides optimum memory performance.

Current best practices call for 1 core-GHz per OSD. We size for 2 GHz CPUs here. Since there are 12 OSDs per server, 6 CPU cores (physical) are needed for the OSDs. As mentioned earlier, separate Optane® (see note) P4800X devices are allocated for BlueStore WAL and RocksDB metadata. Additional CPU resources are needed for driving the Optane® (see note) P4800X devices to provide good performance. Additionally, CPU cores must be available for servicing the operating system and the other Ceph daemons (RGW, MON, and MGR).

Table 5: Sizing CPU physical core requirements

Component	Min. cores per instance	Recommended cores per instance	Instances	Total min. cores	Total recommended cores
Operating system	2	2	1	2	2
Ceph OSD HDD	0.5 (of 2GHz CPU)	1	12	6	12
Ceph MON	1	1	1	1	1
Ceph MGR	1	1	1	1	1
Ceph RGW	3	3	1	3	3
Total	-	-	-	13	19

The table above illustrates that the minimum number of physical CPU cores is 13, with 19 physical cores as the recommended configuration. Although these minimum CPU requirements are quite modest, we chose to specify some additional CPU resources. We did this for four reasons: (1) to accommodate extra load during failure scenarios, (2) to provide additional resources in the event of running multiple RGWs per server, (3) to accommodate caching using the Optane® P4800X drives, and (4) to accommodate Optane® 4800X-based OSDs for bucket indexes. We identified 20 physical cores per storage node to provide ample headroom for these needs. The R740xd is a dual-socket system, allowing the total requirements to be satisfied by 2 CPUs. The **Intel® Xeon® Silver (see note) 4114 CPU** was chosen as it has 10 cores per CPU.

Note: Our performance testing was conducted with P4600 NVMe devices because the P4800X was not orderable at the time the servers were acquired. Please use P4800X instead of P4600.

Note: Our performance testing was conducted with Gold 6130 CPUs because we anticipated experimentation with configurations that require greater CPU resources. Please use Silver 4114 instead of Gold 6130.

Networking

As stated previously, this architecture is based on 25GbE networking components. In accordance with standard Ceph recommendations, two separate networks are used: one for OSD replication, and another for Ceph clients. Standard VLAN tagging is used for traffic isolation. The design includes two **Dell S5248F-ON network switches** to achieve high networking performance and high availability. We recommend two dual port **Intel® XXV710 25GbE NICs** for each storage node. Each network link is made using dual bonded (to achieve 50GbE) connections with each switch handling half of the bond. Similarly, each NIC handles half of a bond. In accordance with common Ceph tuning suggestions, an MTU size of 9000 (jumbo frames) is used throughout the Ceph networks.

Aside from the 50GbE Ceph networks, a separate 1GbE network is established for cluster administration and metrics collection. Additionally, a separate 1GbE network is established for iDRAC access.

Table 6: Storage node networking

Network	NIC	Switch	Description
Ceph cluster	Intel® XXV710 (dual ports)	Dell S5248F-ON	50GbE (dual bonded 25GbE)
Ceph client			
Provisioning, metrics, iDRAC	i350 QP 1GbE NDC, iDRAC embedded	Dell S3048-ON	1GbE

Storage node Ceph NICs

As mentioned earlier, the architecture contains a pair of S5248F-ON switches that are used for both Ceph networks. Additionally, each storage node has a pair of Intel XXV710 25GbE dual port NICs.

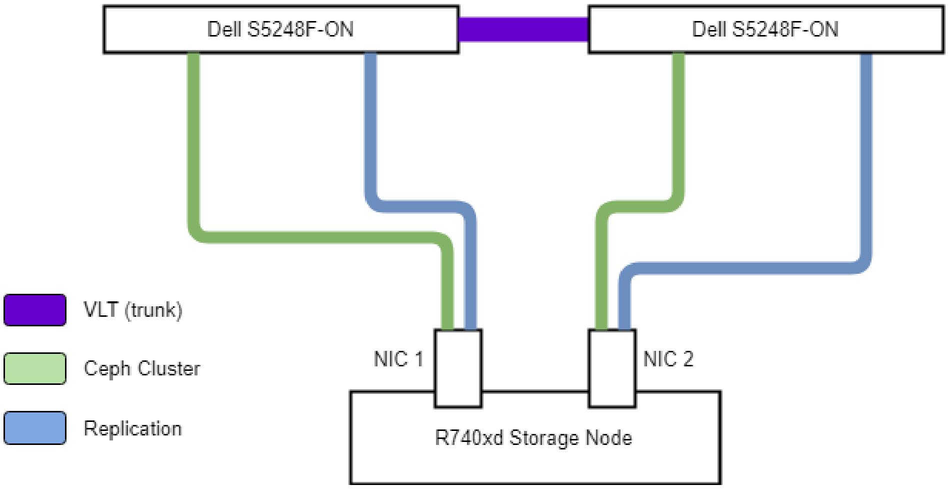


Figure 7: Storage node Ceph networking

The figure above shows how these components are used to integrate the storage nodes into the Ceph networks. As shown in the figure, each network is spread across both switches and both NICs on each storage node. This design provides high availability and can withstand the failure of a NIC, cable, or switch. Additionally, LAG bonds are established for each pair of NIC ports for their respective networks.

Storage node PCIe/NUMA considerations

In order to get the best possible performance, it's necessary to configure devices within the chassis so as to spread the processing load across both CPUs. The two most important device types to separate are the two XXV710 NICs and the two Intel® Optane® add-in cards. Once the XXV710 NICs and the Intel® Optane® add-in cards are separated by NUMA boundaries, the next most important consideration is to separate the HBA330 and the BOSS. The HBA330 is the disk controller that drives all of the SATA HDD devices (OSDs). The BOSS is the embedded RAID1 controller with M.2 storage for the operating system.

Clearly, the HBA330 will carry a far higher I/O load than the BOSS, but there's no need to add unnecessary IO load where it can be avoided. The following figure illustrates how best to balance the devices across the CPUs.

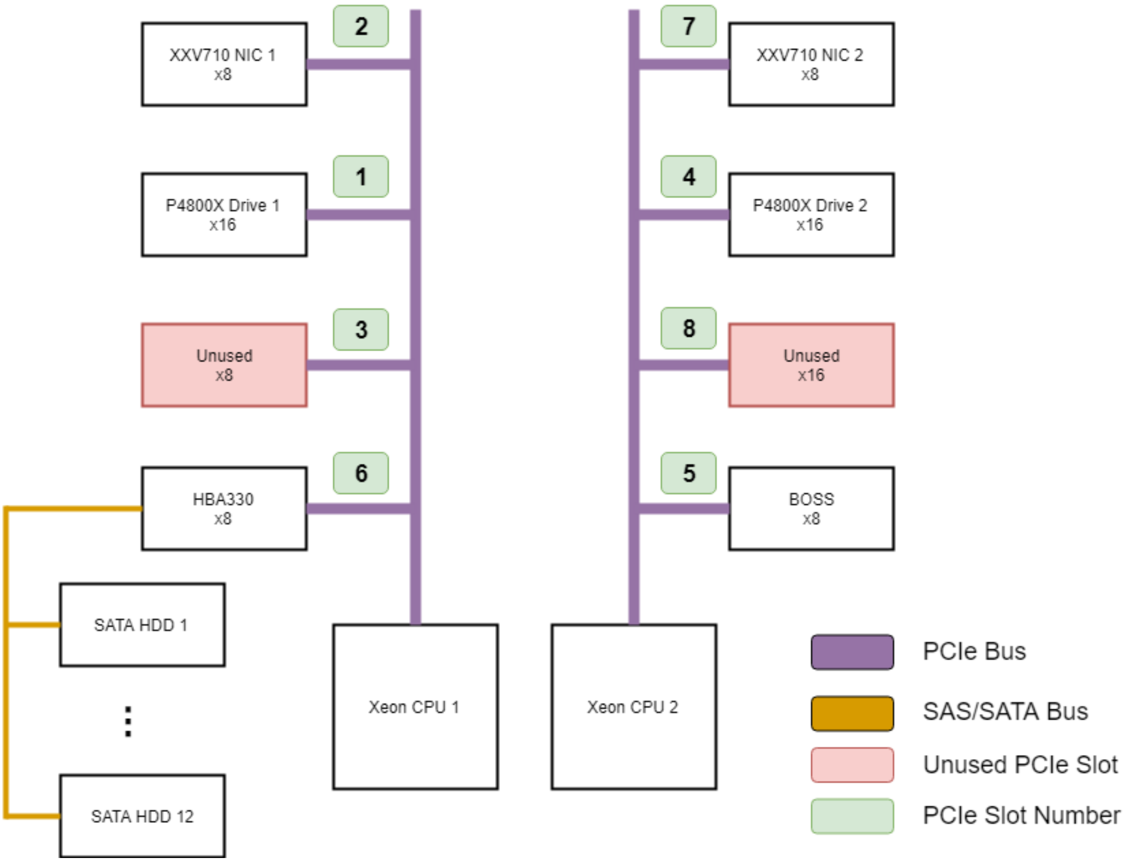


Figure 8: Storage node PCIe slot assignments

The storage nodes used in this architecture make use of Riser Config 6. Each specific riser configuration will have its own set of CPU assignments for PCIe slots. Consulting the specific system diagram is necessary to know these CPU assignments.

Storage node hardware configuration

Table 7: Storage node hardware configuration

Component	Details
Platform	Dell EMC PowerEdge R740xd
CPU	2x Intel® Xeon® Silver (see note) 4114 2.2 GHz
Cores per CPU	10
Memory	192 GB (12x 16GB RDIMM, 2666MT/s)
50GbE (dual bonded 25GbE) network	2x Intel® XXV710 Dual Port 25GbE SFP28
1GbE network	i350 Quad Port 1GbE, rNDC
HDD data storage	12x 10TB (see caution) 7.2K RPM 3.5" SATA HDD 6Gbps

Component	Details
High-speed Ceph metadata storage	2x Intel® Optane® (see note) P4800X 750GB
OS storage	BOSS (2x M.2 Sticks 240G RAID 1)
Disk controller	HBA330 (JBOD)

Note: Our performance testing was conducted with Gold 6130 CPUs because we anticipated experimentation with configurations that require greater CPU resources. Please use Silver 4114 instead of Gold 6130.

CAUTION: Please consult with Red Hat to discuss support for use of drives with capacity above 8TB.

Note: Our performance testing was conducted with P4600 NVMe devices because the P4800X was not orderable at the time the servers were acquired. Please use P4800X instead of P4600.

R640 Admin Node

Aside from the four R740xd storage nodes, a single R640 is included in the architecture to provide the following important functions:

- Collection and analysis of Ceph and server metrics (Prometheus, Grafana, Ceph dashboard)
- Execution of ceph-ansible for deployment and configuration of Ceph on storage nodes
- Administration of all Ceph storage nodes (ssh and iDRAC)

This node is referred to as the **admin node**. The admin node has network connectivity as shown in the following table. This admin node needs connectivity to a single 50GbE (Ceph client) network. As such, it only needs a single dual port 25GbE NIC. The onboard 1GbE NIC is suitable for use with the provisioning and metrics collection network.

Table 8: Admin node networking

Network	NIC	Description
Ceph client	Intel® XXV710 (dual ports)	50GbE (dual bonded 25GbE)
Provisioning, metrics, iDRAC	i350 QP 1GbE NDC, iDRAC embedded	1GbE

Table 9: Admin node hardware configuration

Component	Details
Platform	Dell EMC PowerEdge R640
CPU	2x Intel® Xeon® Gold 6126 2.6 GHz
Memory	192 GB (12x 16GB RDIMM 2666MT/s)
50GbE (dual bonded 25GbE) network	1x Intel® XXV710/2P
1GbE network	i350 QP 1GbE NDC
Storage devices	8x 10K SAS 600 GB HDD
RAID controller	PERC H740P

Note: The memory configuration here uses 12 DIMMs for optimum memory access performance. This is the same reason the R740xd is configured with 12 DIMMs.

Network switches

Our architecture is based on 50GbE (dual bonded 25GbE) networks for core Ceph functionality. Additionally, we establish a 1GbE network for cluster administration, Ceph monitoring, and iDRAC access.

Table 10: Dell network switches

Dell switch configuration		
Dell EMC PowerSwitch S5248F-ON	Cumulus Linux	50GbE Ceph client and replication networks
Dell EMC PowerSwitch S3048-ON	OS9, 48x 1GbE, 4x SFP+ 1GbE	1GbE cluster admin, metrics/ monitoring, iDRAC network

We chose the Dell EMC PowerSwitch S5248F-ON switch as it's the latest and most advanced Dell EMC switch with 25GbE ports and has enough ports to support a full rack of servers. Each S5248F-ON switch contains 48 ports, giving a total of 96 ports for the pair. Each storage node has four 25GbE links (two for each network) with two link connections per switch. This configuration allows the pair of S5248F-ON switches to support up to 24 storage nodes. A standard full-height rack can hold up to 20 storage nodes. Thus, the pair of S5248F-ON switches can handle a full-rack of storage nodes.

Note: Multi-tier networking is required to handle more than 20 storage nodes. Please contact Dell EMC Professional Services for assistance with this more advanced configuration.

Number of storage nodes

Traditionally, a tiny production Ceph cluster required a minimum of seven nodes, three for Ceph MON and at least four for Ceph OSD. The recent ability to deploy colocated, containerized Ceph daemons has significantly reduced these minimum hardware requirements. By deploying Ceph daemons colocated and containerized, one can eliminate the need for three physical servers.

Our architecture consists of a minimum of four storage nodes as part of a minimal starter kit. Since three storage nodes is required for a minimal cluster, having four allows one node to be down due to failure or upgrades. We consider four to be the minimum number of deployed storage nodes. In order to conserve storage capacity, we recommend the use of erasure coding (EC) 4+2 or higher. In order to meet the EC 4+2 configuration in a production deployment, a minimum of seven storage nodes is required. Fewer than seven nodes can be used in the case of 3x replication, but this would require more raw storage capacity to be provisioned. The four storage nodes in our architecture is a starting point for new deployments. The number of storage nodes should be based on your capacity and performance requirements. This architecture is flexible and can scale to multiple racks.

Note: We recommend use of EC 4+2 or higher, along with a minimum of seven storage nodes for production deployments.

Rack component view

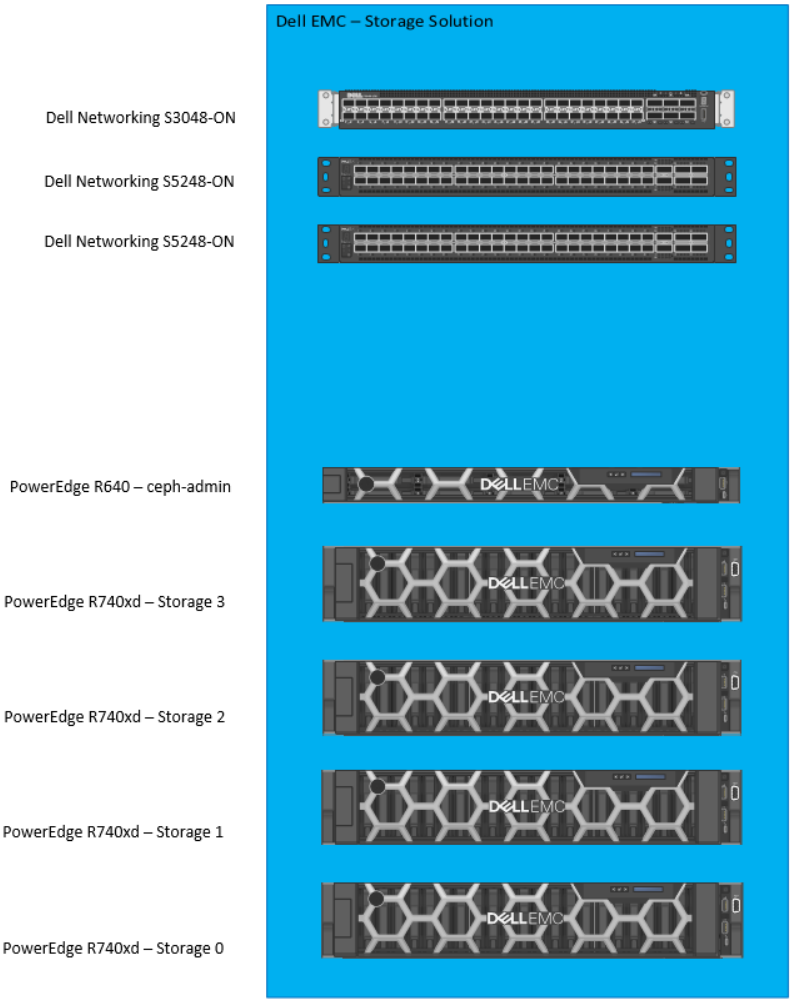

Dell EMC – Storage Solution
Dell Networking S3048-ON
Dell Networking S5248-ON
Dell Networking S5248-ON
PowerEdge R640 – ceph-admin
PowerEdge R740xd – Storage 3
PowerEdge R740xd – Storage 2
PowerEdge R740xd – Storage 1
PowerEdge R740xd – Storage 0

Figure 9: The 4-Node Ceph cluster and admin node based on Dell PowerEdge R740xd and R640 servers

Note: We recommend that heavier storage nodes be located at the bottom of the rack.

Note: We use four storage nodes as a starting point. You should consider seven or more for production.

The R740xd is a 2U server, while the R640 and the switches are each 1U. Taken as a whole, the cluster of four storage nodes and one admin node requires 9U for servers and 3U for switches.

It is worth noting that if the MON daemons were not containerized and colocated (with OSD and RGW), the rack space requirements would be increased by 3U (e.g., 3 R640). This deployment topology provides significant cost savings and noticeable rack space savings.

Software

Table 11: Architecture software components/configuration

Component	Details
Operating system	Red Hat Enterprise Linux (RHEL) 7.6
Ceph	Red Hat Ceph Storage (RHCS) 3.2
OSD backend	BlueStore
CPU logical cores per OSD container	3
CPU logical cores per RGW container	18
Ceph storage protection	Erasure coding 2+1 (see note)
Ceph daemon deployment	Containerized and colocated

Note: We used a 2+1 EC profile due to only having four nodes. Production deployments using EC should use 4+2 or higher with a minimum of seven storage nodes.

Architecture summary

The architecture presented in this chapter was designed to meet specific objectives. The following table summarizes how the objectives are met.

Table 12: Architecture objectives

Objective	How met
High capacity	• Use of high capacity 3.5" drives • Use of erasure coding
Very good performance	• Xeon® Silver (see note) 4114 CPUs (10C) • 50GbE (dual bonded 25GbE) networking • Optane® (see note) P4800X high speed storage devices • Separate replication network • Jumbo frames enabled
Cost effective	• Colocated daemons (reduce server count) • SATA HDD (most cost effective for high capacity) • 25GbE networking components (sweet spot for cost/performance)
High availability	• Redundant 25GbE switches • Redundant 25GbE NICs • Hot-plug storage devices • Redundant power supplies
Leverage Ceph 3.2 improvements	• BlueStore OSD backend • Containerized daemons

Objective	How met
Easy to administer	• iDRAC9 remote server admin • Separate, integrated Ceph admin node • Ceph dashboard (Grafana based)

Note: Our performance testing was conducted with Gold 6130 CPUs because we anticipated experimentation with configurations that require greater CPU resources. Please use Silver 4114 instead of Gold 6130.

Note: Our performance testing was conducted with P4600 NVMe devices because the P4800X was not orderable at the time the servers were acquired. Please use P4800X instead of P4600.

Chapter 4

Test setup

Topics:

This chapter highlights the procedure used to setup the storage cluster along with other components. It includes physical setup, configuration of servers, and deployment of RHEL and RHCS.

Physical setup

The equipment was installed as shown below. When installing the 25GbE NICs in the servers, care needs to be taken to ensure the cards are on separate NUMA nodes. This ensures that the traffic is handled by different CPUs for individual NICs and the traffic is spread across the CPUs.

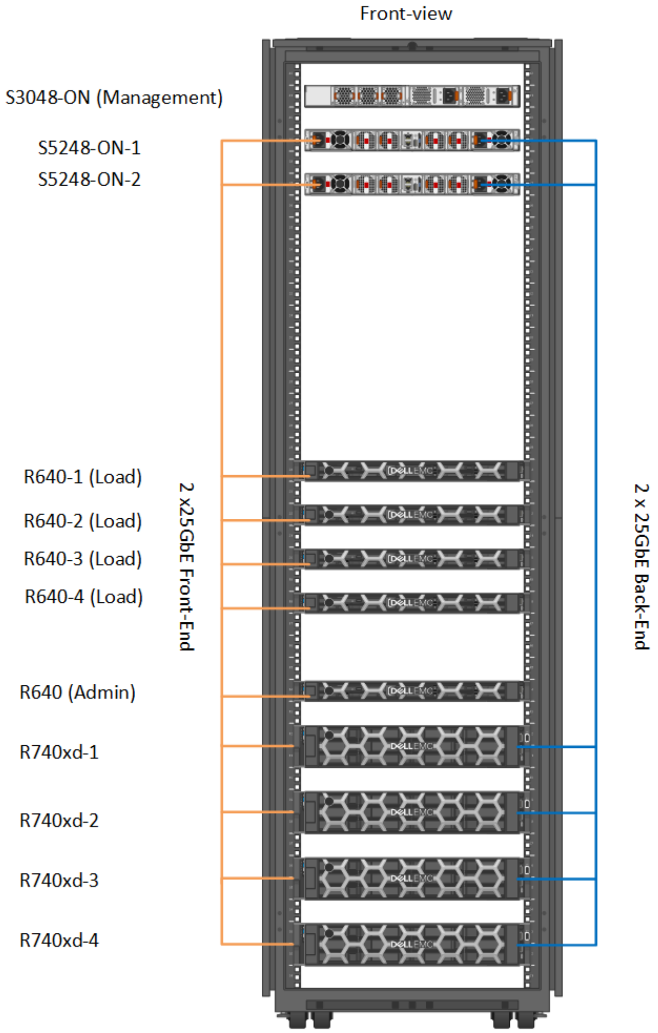


Figure 10: Ceph cluster with R640 Servers as Load generators

Note: We recommend that heavier storage nodes be located at the bottom of the rack.

Note: We use four storage nodes as a starting point. You should consider seven or more for production.

Each Ceph storage node has four 25GbE links going into two leaf switches. These links are created keeping the high availability and bandwidth aggregation architecture in context. One set of the 25GbE links (as an 802.3ad LAG, or bond in terms of RHEL) is connected to the frontend (ceph-client/public API) network. The other set of links is connected to the backend (ceph-storage) network. The load generator servers have 2 x 25GbE link connected to the frontend network. A separate 1GbE management network is used for administrative access to all nodes through SSH.

Note: The following bonding options were used: mode=802.3ad miimon=100 xmit_hash_policy=layer3+4 lacp_rate=1

While the overall physical setup, server types, and number of systems remain unchanged, the configuration of the OSD node's storage subsystems was altered. Throughout the benchmark tests, different I/O subsystem configurations are used to determine the best performing configuration for a specific usage scenario.

Table 13: Software components in testbed

Software components in testbed	
Ceph	Red Hat Ceph Storage 3.2
Operating system	Red Hat Enterprise Linux 7.6
Tools	COSBench
Server monitoring	Prometheus (node exporter) and Grafana

Table 14: Ceph configuration used in all benchmarks

Configuration	Details
Erasure coding	2+1 (see note)
Number OSDs (cluster-wide)	48 (1 per SATA HDD)
Number RGWs (cluster-wide)	4 (1 per storage node)
Ceph Write Journal Devices	2 NVMe (see note) devices serving 12 OSDs (6:1)

Note: We used a 2+1 EC profile due to only having four nodes. Production deployments using EC should use 4+2 or higher with a minimum of seven storage nodes.

Note: Our performance testing used NVMe devices because P4800X devices were not available for order when our servers were acquired.

Configuring Dell PowerEdge servers

The Dell PowerEdge R740xd and Dell PowerEdge R640 servers are configured using the iDRAC and the racadm configuration utility. The iDRAC configuration is deployed on the admin node and used to reset the server configuration, including the BIOS configuration. This ensures all systems have the same configuration and were set back to known states between configuration changes. With the `racadm` command, the configuration can be retrieved from and stored to an NFS share, which is provided by the admin node.

Deploying Red Hat Enterprise Linux

To deploy RHEL, the recommended approach is through a coordinated and centralized installation server. This not only reduces the deployment time significantly but also improves the consistency in configurations (for example network configs) by avoiding the manual error-prone setup on individual servers. In our setup, we deploy RHEL 7.6 on all the nodes. This includes the administration node, which handles RHEL as well as RHCS installation, test log aggregation, monitoring, and other management related tasks. This node will be referenced as admin node through the remainder of this document.

Table 15: Required services

Service	Notes
NTP	Time synchronization is very important for all Ceph nodes
DNS	Not strictly required for Ceph, but needed for proper RHEL functioning

Deploying Red Hat Ceph Storage

In production environments, Red Hat Ceph Storage can be deployed with an easy-to-use Ansible playbook, ceph-ansible.

Ceph-ansible is an end-to-end automated installation routine for Ceph clusters based on the Ansible automation framework. Predefined Ansible host groups exist to denote certain servers according to their function in the Ceph cluster, namely OSD, MON, MGR, and RGW nodes. Tied to the predefined host groups are predefined Ansible roles. The Ansible roles are a way to organize Ansible playbooks according to the standard Ansible templating framework, which in turn, are modeled closely to roles that a server can have in a Ceph cluster.

Note: We recommend running ceph-ansible from the admin node. It provides adequate network isolation and ease of management.

The Ceph daemons are colocated as containerized services when deployed. Specifically, out of the four nodes in the architecture, since MONs are to be deployed in an odd number (to maintain consensus on cluster state), they're deployed on three out of four nodes, whereas OSD and RGW daemons are deployed on all four nodes. This is shown in the figure below. However, these daemons are isolated on the OS level through use of Linux containers. Since three MONs are enough to maintain cluster state, additional storage nodes (with OSD and RGW only) can be added to expand the cluster by simply plugging the additional hardware and deploying OSDs and RGWs on the nodes.

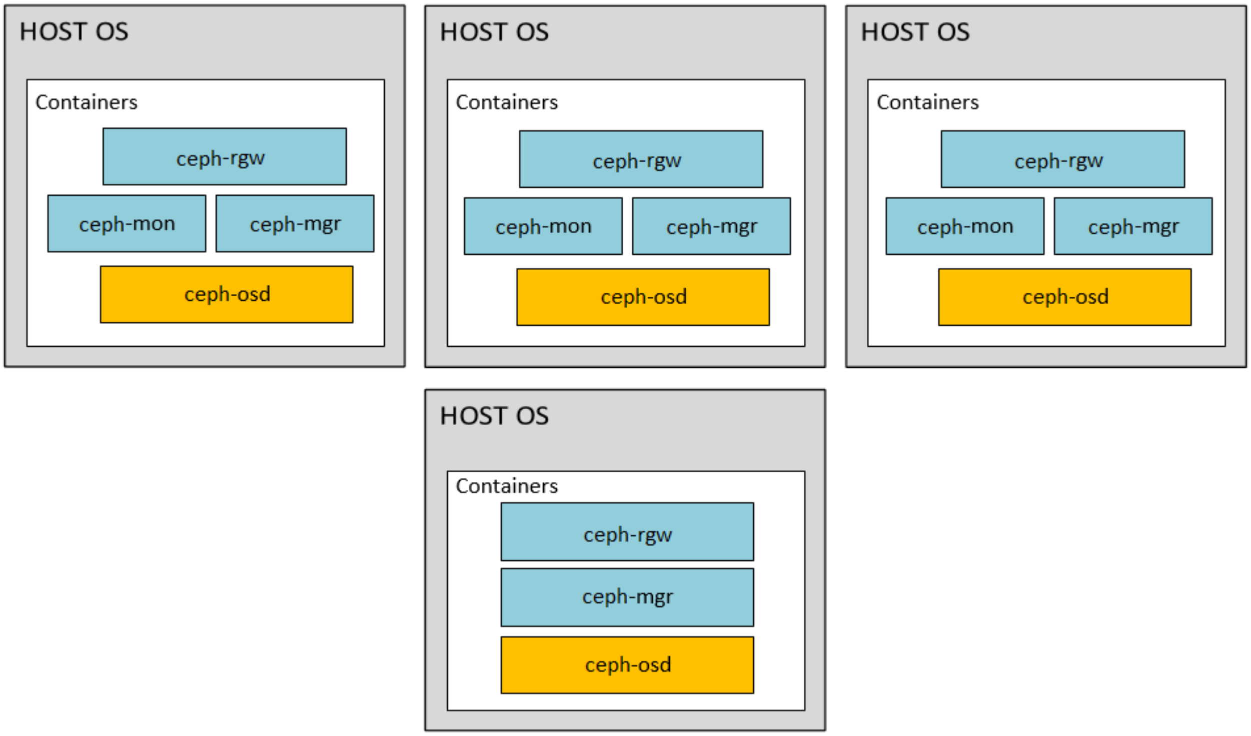


Figure 11: Colocated containerized Ceph block storage

Table 16: Deploying Ceph daemons

Ceph daemon	Deployment
MON	3 for small to mid-sized clusters, 5 for large clusters
MGR	Same as that for MON; put MGR on same nodes as MON
OSD	12 per storage node (1 per HDD)
RGW	1 per storage node; load balancer single endpoint for clients

Metrics collection

In order to pinpoint bottlenecks encountered during testing, it's critical to have a variety of server metrics captured during test execution. We made use of Prometheus for monitoring our servers. We installed the standard 'node exporter' on each node and configured our Prometheus server to pull the server metrics every 10 seconds.

The 'node exporter' captures all of the standard server metrics that are used for analysis. Metrics are captured for network, CPU, memory, and storage devices. Our Prometheus server was integrated with Grafana to provide a rich and powerful monitoring tool. This infrastructure allowed to us seamlessly capture all relevant metrics during all of our test runs.

Test and production environments compared

Table 17: Differences between performance testing and production environments

Area of difference	Performance testing	Production
S3 client connectivity	Each load generator uses hard-coded, specific RGW instance (no load balancer)	Single endpoint (load balancer)
COSBench load generators	present	N/A
Erasure coding (EC)	2+1	4+2 (or higher)
Number storage nodes	4	7 (or higher)
Ceph authentication	none (disabled)	cephx (enabled)
Ceph scrubbing	disabled	enabled
Ceph data CRC checks	disabled	enabled
Vulnerability patches	disabled	enabled
High-speed Ceph metadata device	Intel® P4600 NVMe	Intel® Optane® P4800X
CPU model	Intel® Xeon® Gold 6130	Intel® Xeon® Silver 4114

Note: We avoid use of a load balancer in performance testing to eliminate possibility of load balancer being a bottleneck.

Chapter 5

Test methodology

Topics:

This chapter details the innovative tools and workload testing methodology yielding expedient, robust, and unique testing cycle configuration.

Overview

The methodology used in the testing process was designed in a way that allows us to view the cluster behavior under various conditions and workloads. We perform write as well as read operations on the gateway. We make use of the S3 API and the object sizes used are 64KB, 1MB, and 4MB. For each object size, we test the cluster with both 100% write and 100% read workloads for a selected range of clients. Similarly, we choose an EC profile of 2+1 due to limited number of nodes. This gives us a rich insight into the cluster behavior, because it allows extrapolation and speculation for numerous production environment workloads.

The first stage is to prefill the cluster to avoid testing against empty disk drives. We need this for realistic numbers. Then we prepare all the necessary data needed for benchmarking, and finally we run the whole testing suite and retrieve results through scripts.

For each test and every object size, we then record the average RADOS Gateway throughput as well as the average latency (response time). We gather these numbers from COSBench results and comparison with Grafana/Prometheus numbers shows consistency. This allows us to perform sanity testing on numbers.

We developed a custom benchmarking methodology that allowed us to avoid a lot of overhead in conventional COSBench testing cycles. This also assisted automation and sequential running of a complete suite of tests and allowed for cache clearing as well.

Workload generation

All of our workloads were generated using COSBench (*discussed later*).

The first step is to sufficiently prefill the cluster so that the HDDs have enough data to ensure realistic seek times. Without an adequate prefill of data the tests results are likely to be overstated (unrealistically high). Therefore, we fill the cluster to approximately 40% of it's capacity. This prefill is governed by parameters listed in the following table. Other methodology constants are also listed.

Table 18: Methodology constants

Category	Parameter	Value
Pre-populated data	Number containers	400
	Objects per container	10,000
	Total objects	4,000,000
	Object size	32MB
	Total data filled	128TB (~40% of cluster capacity)
Data durability	Erasure coding	k=2, m=1
Client connectivity	Object API	S3
Cluster endpoints	RGW - no load balancer; load generator worker has fixed RGW endpoint	4 RGW instances (1 per storage node)
Load generation	Number nodes	4 (1 per storage node to maintain 1:1)

After the prefill has completed (about 16 hours in our tests), the next step is to prepare all the data needed for testing. In a conventional COSBench workload, every workload begins with init/prepare stages, where containers are created and objects are populated. This is followed by the read/write workload stages as required and finally, the cleanup/dispose stages delete the objects and containers prepared earlier. It takes a significant amount of time to prepare the objects needed, and takes equally long to cleanup at the end of the workload. Therefore, we redesigned the methodology to reduce the test cycle time.

We create init/prepare stage based workloads and filled the cluster as required (for each object size) only once (without a need for cleanup as well). This not only cut down on hours of overhead in each workload, but also made the testing process easier to manage, since all that was needed now was the main testing stages, i.e. read/write. This data preparation is summarized below.

Table 19: Testing data preparation parameters

Object size	Containers used	Objects per container	Total objects	Total data filled
64KB	400	90,000	36,000,000	2.3TB
1MB	200	10,000	2,000,000	2TB
4MB	200	2,500	500,000	2TB

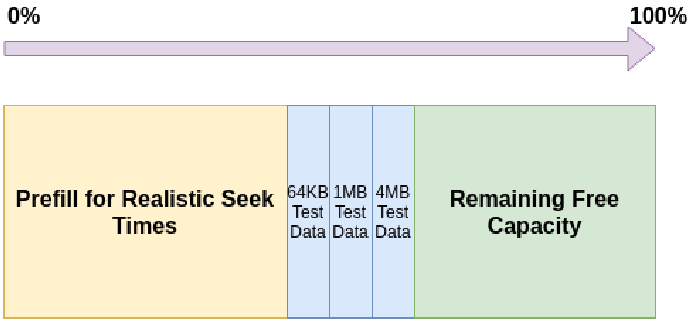


Figure 12: Illustration of cluster filling

Note: The total amount of data needed to prepare and subsequently use in testing is significantly larger than the total amount of RAM in the cluster. In our case, with 784GB RAM, 2TB is approximately 2.5x the amount of RAM. This is to reduce caching effects to a minimum.

Table 20: Methodology variables

Category	Parameter	Values
Payload	Object sizes	64KB, 1MB, 4MB
Cluster loading	Concurrent clients	40, 80, 120, 160, 200, 240, 280
Workload	S3 operations	100% write, 100% read

COSBench

COSBench, an open-source benchmarking tool, was used to measure Cloud Object Storage service performance. It is developed by Intel. In object storage services, objects are placed in buckets. For workload generation, we simulate diverse usage patterns from workload models defined upon the storage interface.

There are two components in COSBench: controller and driver. Both components are required while running the experiment. The controller is added to supervise multiple drivers in a distributed environment so that they can work collaboratively. The results are aggregated on the controller node. The driver node runs test driver processes and can host multiple driver processes.

Each driver process can then execute workload with threads known as workers. Optimum load generation is performed when the number of workers is a multiple of the number of driver processes. In our case, we have 20 driver processes running on four load generator nodes, and therefore, we choose workers (number of clients) as a multiple of this number.

Each workload is written as an XML file with separate stages defined in separate tags. COSBench also provides a CLI interface, which is ideal for automation. We used this CLI to submit customized workloads through scripts. The results can then be collected easily from the archive directory.

In the figure below, the head node corresponds to the controller node, which has the controller process running, and the client nodes are the driver nodes which have driver processes running. The figure depicts the typical test execution cycle of COSBench on an RHCS object storage cluster.

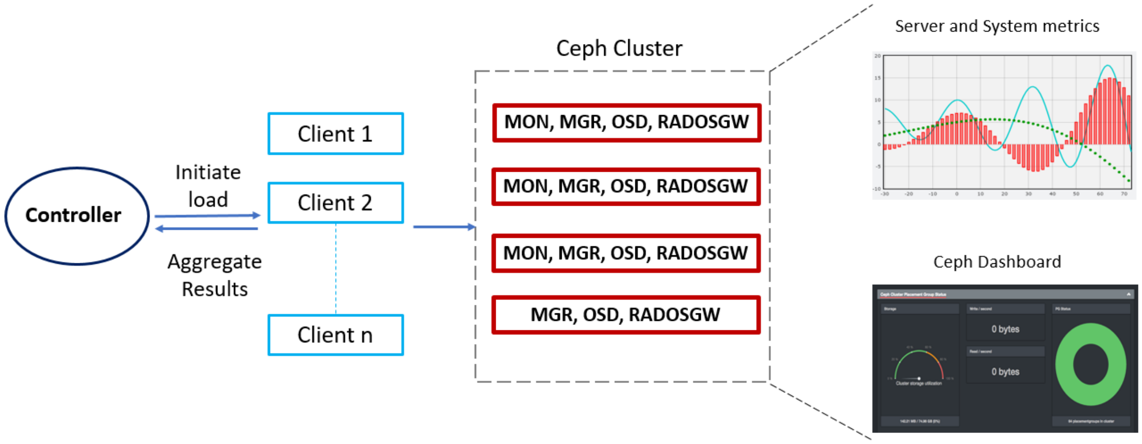


Figure 13: Test methodology components

Iterative tuning

A baseline configuration was established before commencing the baseline benchmarks. This baseline configuration largely consisted of default values, but with a few deliberate changes. For example, we knew ahead of time that our tests would be conducted with Ceph authentication (cephx) turned off. Consequently, we turned it off from the very beginning. We selected a starting value of 2048 placement groups (PGs) for our baseline configuration. Once our baseline configuration was established, we ran the predefined workloads on our cluster with increasing load from our load generators.

For object storage, Ceph requires a number of pools to be created in the cluster. The pools critical to performance are bucket data and bucket index pools. The PGs mentioned are for bucket data pool, since that stores ~99.9% of the cluster data. The index pool is a 2x replication pool and is mapped on the NVMe OSDs with 64 PGs. This mapping of the pool on NVMe OSD is the only optimization necessary from performance perspective.

Once a baseline set of metrics were obtained, a series of iterative tuning efforts were made in an effort to find the optimal performance. The first step was to ensure correct hardware placement on each node (*see "Storage node PCIe/NUMA considerations"*). Secondly, we used an LVM based deployment, to allow us to use the NVMe drives as OSDs and cache drives simultaneously. We then placed the bucket index pool on the NVMe OSDs using CRUSH placement rules for increased performance. This is a widely used configuration in the Ceph community for object storage clusters.

We chose an EC profile of 2+1 since we have four nodes in our cluster. In production environments, however, with seven or more nodes, an EC profile of 4+2 or higher is recommended. The choice depends on the number of nodes in the cluster. For n nodes, n=k+m+1 where k and m are data and code chunks of EC profile. Therefore, with our four node cluster, (4=2+1+1) 2+1 was the available profile for us.

The final consideration is resource sizing for containers of different services. We've determined these values through experimentation while monitoring the resource utilization of containers during heavy load. The resource requirements are discussed in *"CPU and memory sizing"*. Here, we reiterate that adequate resource assignment to containers is a critical factor in performance tuning, and significant performance degradation can be observed with improper resource assignment to containers. The procedure to determine these can be starting with reasonable values and exploring nearby values in steps, until an optimum value has been obtained.

⚠ **CAUTION:** Some tuning options have been set to obtain maximum performance and to enable like-comparison with other published whitepapers. These tuning options are not recommended for production use.

Table 21: Tuning with caution

Area	Config. option	Performance	Production
Ceph	auth_client_required	none	cephx
	ms_crc_data	False	True
	scrubbing	disabled	enabled
OS	Meltdown/spectre/ ZombieLoad patches	disabled	enabled

Testing approach

As discussed previously *(see "Workload generation")*, the cluster was first prefilled to a significant capacity to ensure realistic seek times. Then all the data needed for benchmarking was prepared on the cluster to avoid prepare/cleanup cycle overheads. Each test had two iterations to ensure consistency of numbers. The reported numbers are the average of those two test iterations.

For read workloads, we performed a series of experiments to determine the relationship between the workloads and OSD cache memory. We observed that the performance remained consistent throughout successive test runs, and it wasn't beneficial to drop caches. This is not unordinary behavior, if you consider that Bluestore, unlike legacy Filestore does not rely on file system for its metadata.

However, for write operations, caches were dropped between every iteration of every test, since these tests heavily rely on OSD cache memory. The overall methodology is summarized below.

Table 22: Methodology summary

Methodology stage #	Stage description
1	Pre-populate data
2	Prepare benchmark data
3	Run benchmark tests for every object size

To drop the OSD caches between successive iterations, the conventional method of dropping caches in Linux ("echo 3 > /proc/sys/vm/drop_caches") cannot be used. This is because BlueStore OSD devices do not store data in a file system. The data is stored directly on the block device in unstructured form. Therefore, all the OSD daemons need to be restarted on every node to clear out the cache. After the OSDs are restarted, 5 to 6 minutes of ramp time is given to let the cluster reach steady state. Once the cluster reaches steady state, testing can continue. The exact details of this method are provided in the appendix.

Finally, two very important functional components of Ceph were disabled during all tests. These components relate to data integrity checks and security.

Scrubbing is a critical function of Ceph used to verify the integrity of data stored on disk. However, scrubbing operations are resource intensive and can interfere with performance. We disable scrubbing to prevent adverse performance effects and to enable study in a more controlled and predictable manner. Scrubbing should not be disabled in production.

⚠ **CAUTION:** Although we disable scrubbing during controlled performance tests, scrubbing must not be disabled in production! It provides a critical data integrity function.

Cephx is an important security function that provides Ceph client authentication. This feature is enabled by default and is recommended for all production use. We disable it in our tests since this feature is commonly disabled in other Ceph benchmarks. This makes our results more comparable with other published studies.

⚠ **CAUTION:** We disabled Cephx for performance studies, but this feature should not be disabled in production! It provides a critical data security function.

Chapter 6

Hardware baseline testing

Topics:

This chapter presents our hardware baseline tests performed on various components such as CPU, network, and storage devices.

Baseline testing overview

Before attempting benchmark scenarios that utilize higher-layer Ceph protocols, it is recommended to establish a known performance baseline of all relevant subsystems. We perform hardware baseline tests for CPU, network, and storage devices.

CPU baseline testing

CPU testing has been performed with Intel® LINPACK benchmark running suitable problem sizes given each server's CPU resources.

Table 23: CPU baseline

Server type	**Storage node:** **PowerEdge R740xd 2x Intel® Xeon® Gold 6130 CPU @ 2.10GHz**	**Load generators:** **PowerEdge R640 2x Intel® Xeon® Gold 6126 CPU @ 2.60GHz**
LINPACK results	443 GFlops (problem size = 30000)(cores = 64)	448 GFlops (problem size = 30000)(cores = 48)

Network baseline testing

Network performance measurements have been taken by running point-to-point connection tests following a fully-meshed approach; that is, each server's connection has been tested towards each available endpoint of the other servers. The tests were run one by one and thus do not include measuring the switch backplane's combined throughput, although the physical line rate is 50000 MBit/s for each individual link. An MTU value of 9000 was used throughout all tests.

Note: The iPerf network performance utility was used to establish networking baselines.

Table 24: Network baseline

Server type	**PowerEdge R740xd Intel® XXV710**	**PowerEdge R640 Intel® XXV710**
PowerEdge R740xd Intel® XXV710	46.43 GBit/s	46.43 GBit/s
PowerEdge R640 Intel® XXV710	46.42 GBit/s	46.42 GBit/s

Storage device baseline testing

The drive performance measurements were taken with FIO using libaio backend. The test profile used a 4MB block size, one thread for both sequential read and write, a queue depth of 32 and direct I/O (no page cache). Even for parallel drive operations, a single thread was enough to stress all the drives in parallel. Each test had a runtime of 5 minutes. The objective was to determine the practical bounds of metrics, which were then to serve as points of comparison in the performance tuning process.

Table 25: NVMe baseline - sequential read

# Drives	Total throughput (MB/s)	Throughput per drive
1	3357	3357
2	6711	3355.5

Table 26: NVMe baseline - sequential write

# Drives	Total throughput (MB/s)	Throughput per drive
1	1657	1657
2	3389	1694.5

Note: Our performance testing used NVMe devices because P4800X devices were not available for order when our servers were acquired.

For the SATA HDD measurements, one thread was used, even for multiple drives. The rest of the profile was the same as in the case of NVMe SSDs. Notice that the throughput doesn't scale linearly with the number of drives. This is expected with HDDs.

Table 27: SATA HDD baseline - sequential read

# Drives	Total throughput (MB/s)	Throughput per drive
1	250	250
12	2990	249.1

Table 28: SATA HDD baseline - sequential write

# Drives	Total throughput (MB/s)	Throughput per drive
1	195	195
12	1498	124.8

Chapter 7

Benchmark test results

Topics:

This chapter provides the benchmark results along with the bottleneck analysis. The bottleneck analysis is conducted using hardware usage metrics that were gathered throughout the testing process. Finally, it summarizes the key takeaways from the performance analysis work.

Note: Our performance testing used NVMe devices because P4800X devices were not available for order when our servers were acquired.

Bottleneck analysis

In previous chapters, we discussed *(see "Metrics collection")* our Prometheus and Grafana server monitoring infrastructure. This infrastructure captured all relevant metrics related to CPU, memory, OS, networking, and storage devices. We were then able to analyze these various server metrics from the same time period when specific tests were run.

This analysis enabled us to identify bottlenecks that were hit within various benchmark tests. The most important metrics for this analysis included:

- CPU utilization
- Network throughput
- Memory utilization
- Storage device utilization
- Storage device throughput

All the metrics presented for analysis are from the Ceph storage nodes. This includes the CPU utilization, network utilization, and drive utilization numbers.

64KB read

The test was run on the data prepared as discussed earlier in the methodology chapter. We see an increase in performance corresponding to increasing number of clients up to 120 clients. This is where the bandwidth begins to saturate and latency delta increases. The network bandwidth as well as CPU and memory usages are well within bounds. The identified bottleneck is the HDDs.

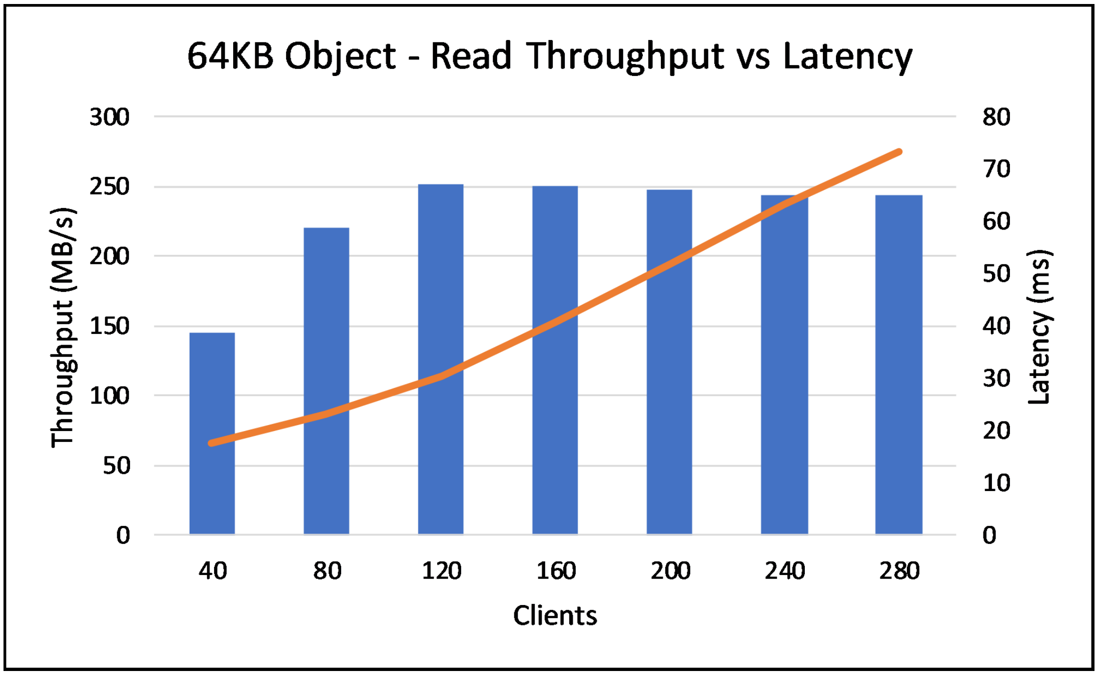


Figure 14: 64KB Object Read Throughput

The metrics shown below are for both test iterations. The drives are at a very high utilization and therefore it is clear that they are the bottleneck. This is expected since we have a large number of smaller transactions and because of frequent seek time overheads. The multiple peaks depict the two iterations of the test, to show repeatability of numbers.

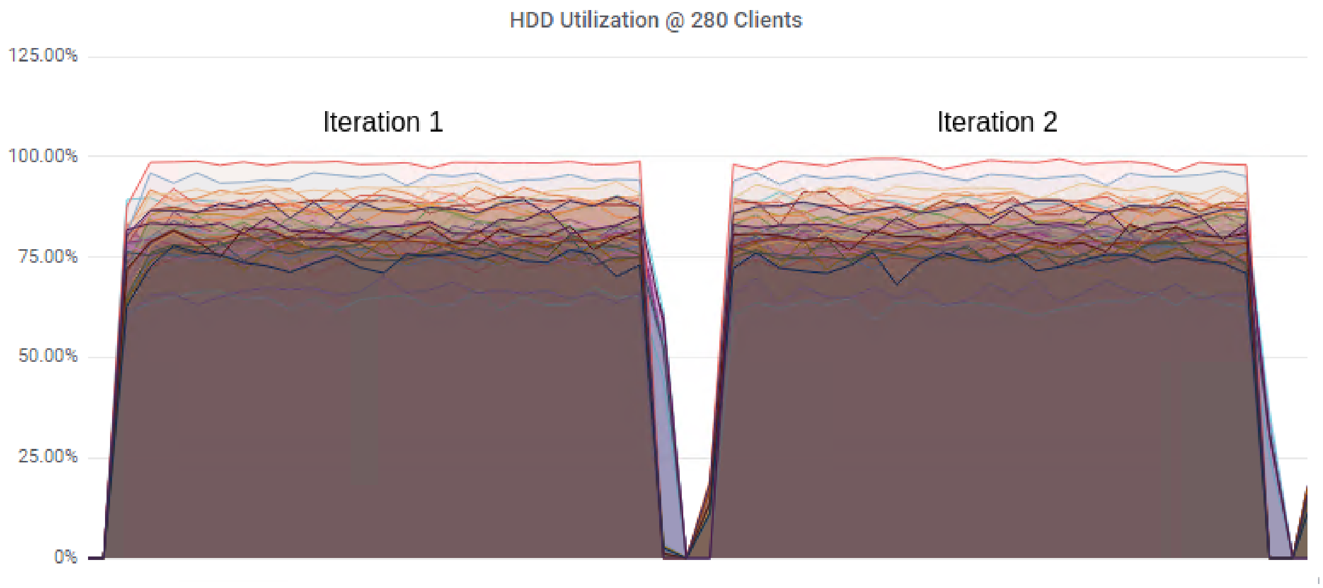


Figure 15: 64KB Object HDD Utilization

64KB write

The prepared data was used to run the write workload as described earlier. The performance scales well until 160 clients where the throughput starts to level off. This is the point of interest for analysis. The CPU, memory and network usage metrics were all indicated well within bounds. The BlueStore NVMe devices were also operating well below their expected bounds. The HDDs were again the bottleneck in this case.

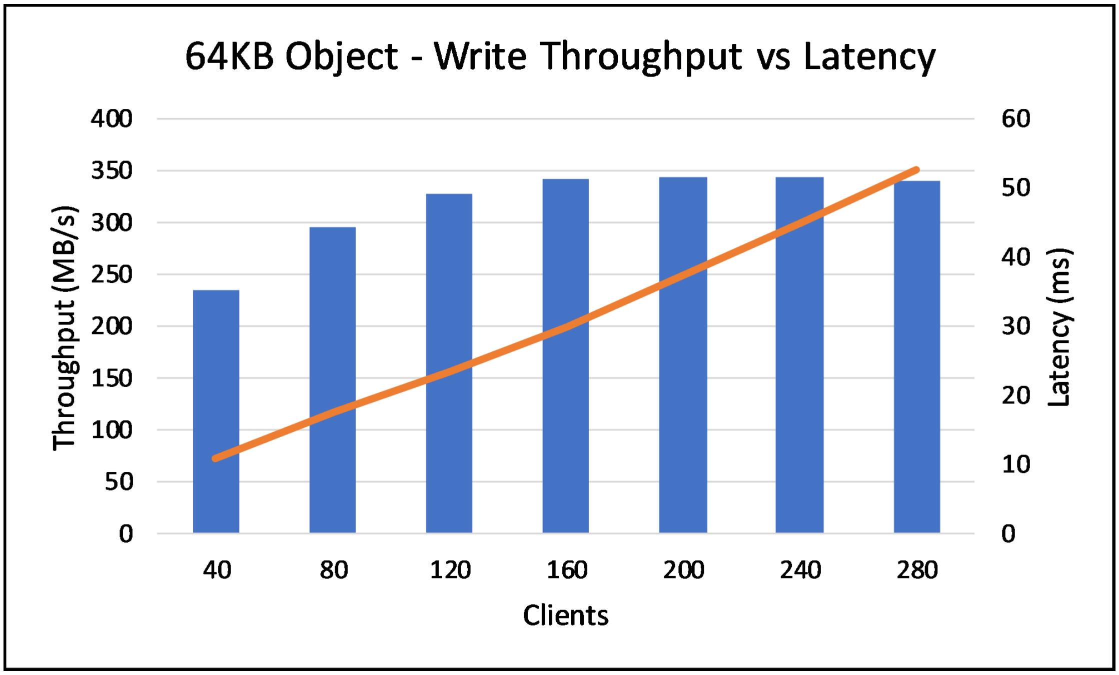


Figure 16: 64KB Object Write Throughput

As indicated in the metrics below, it is clear that the drives are overloaded in their seek times and are operating in a saturated state. This is again because of the workload composed completely of very small transactions that require too much seek time overhead.

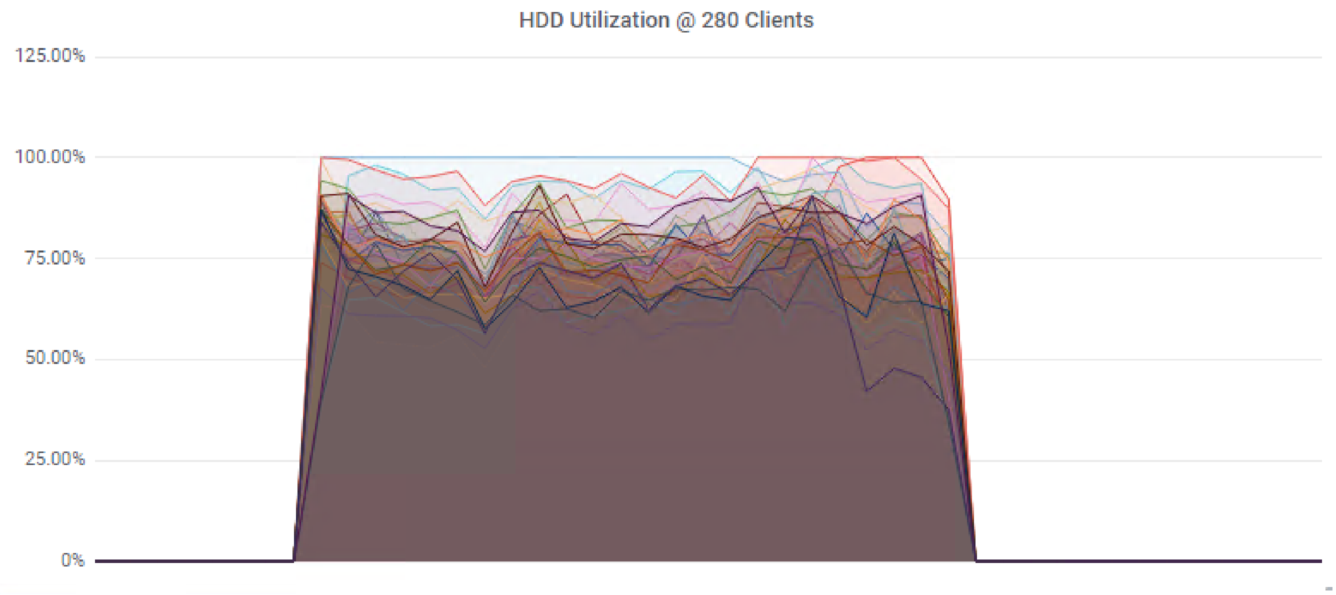


Figure 17: 64KB Object HDD Utilization

1MB read

This workload was again executed on prepared data. The performance scales until 120 clients where the throughput saturates. The identified bottlenecks were hard drives yet again. However, since this workload is more effective, since it is utilizing not only the seek time capacity but also the throughput capacity of drives, unlike reads of smaller objects (64KB for example).

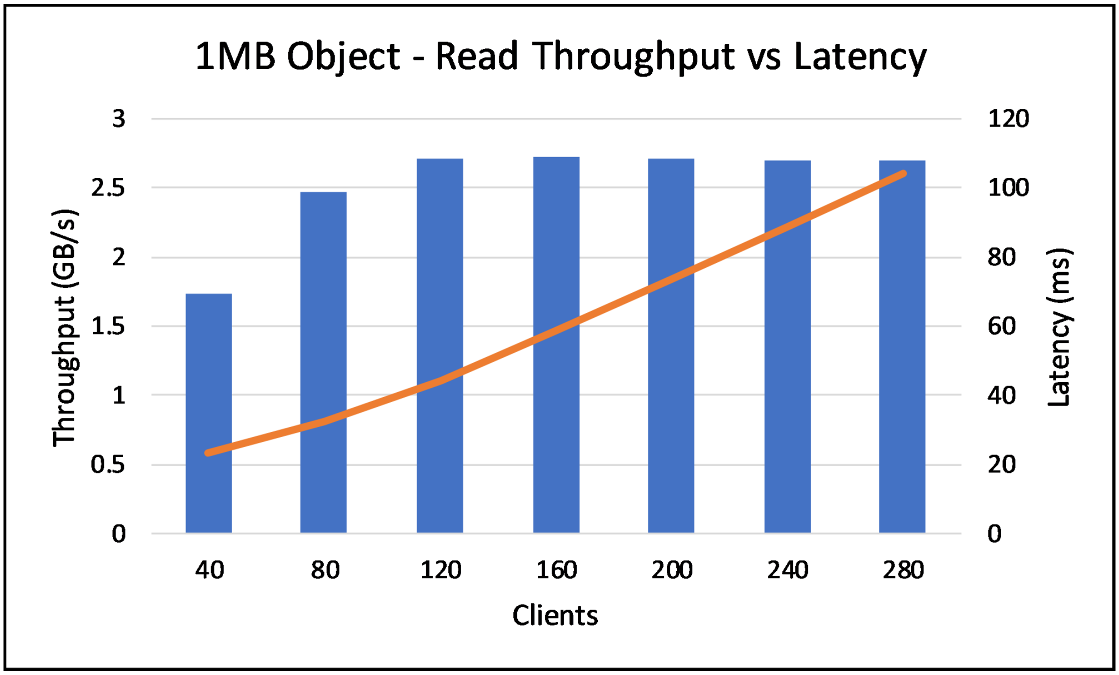


Figure 18: 1MB Object Read Throughput

The HDDs are saturated on their read bandwidth due to a large object size. Also notice that the utilization numbers shown below are lower than those of 64KB object. This is expected, since larger object transactions imply fewer transactions (remember, total data processed is same in all object sizes) and hence less impact of seek latencies.

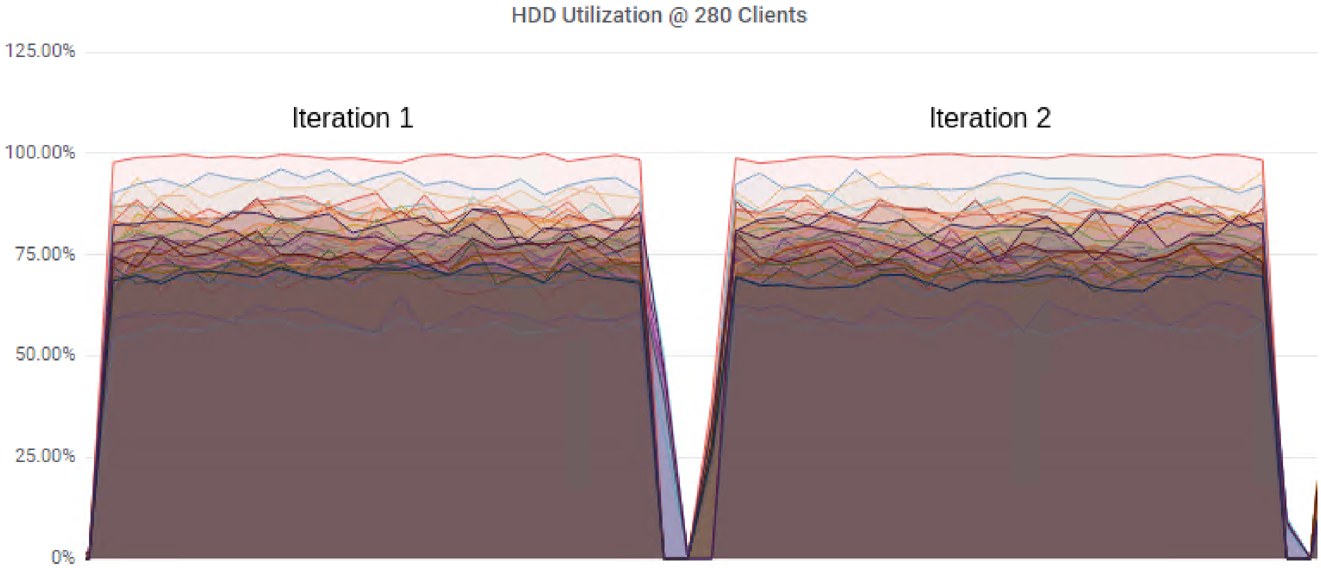


Figure 19: 1MB Object HDD Utilization

1MB write

The throughput scaled very well up to 160 clients, where the scaling slowed significantly. This is the point of interest for analysis and hence the drive utilization metrics are shown below. Once again, all other system resources (CPU, memory, network) were not the cause of system throughput saturation.

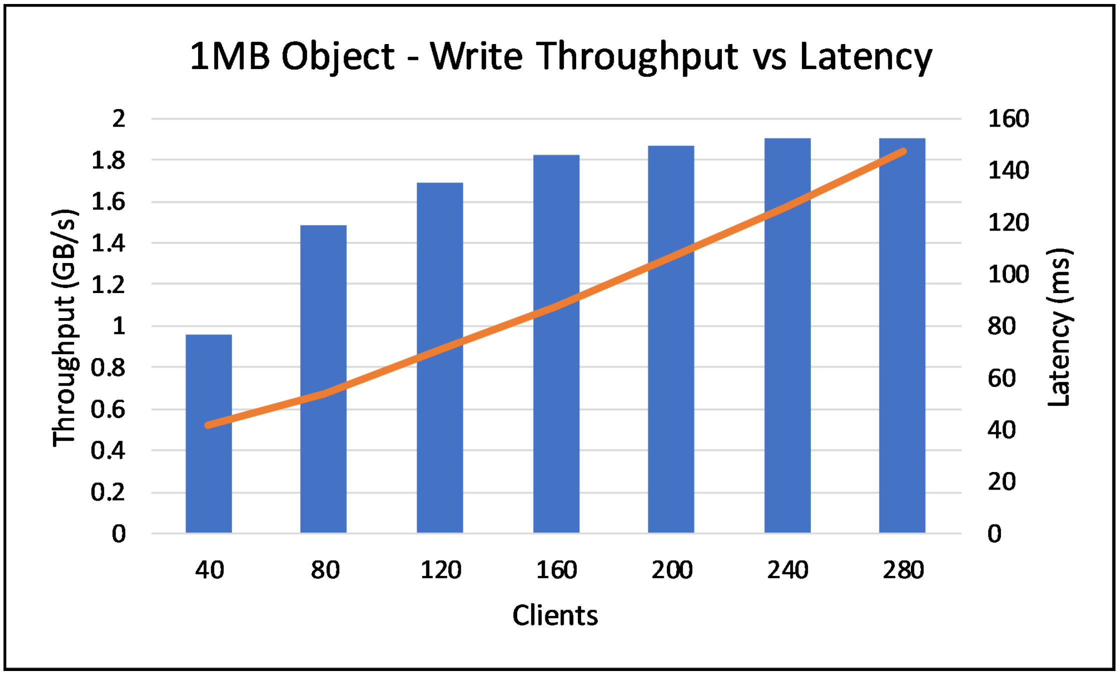


Figure 20: 1MB Object Write Throughput

It is clear from the graph below that this workload utilizes the drives very effectively. They saturate not only on bandwidth, but also on I/O time and are servicing the workload at their full capacity.

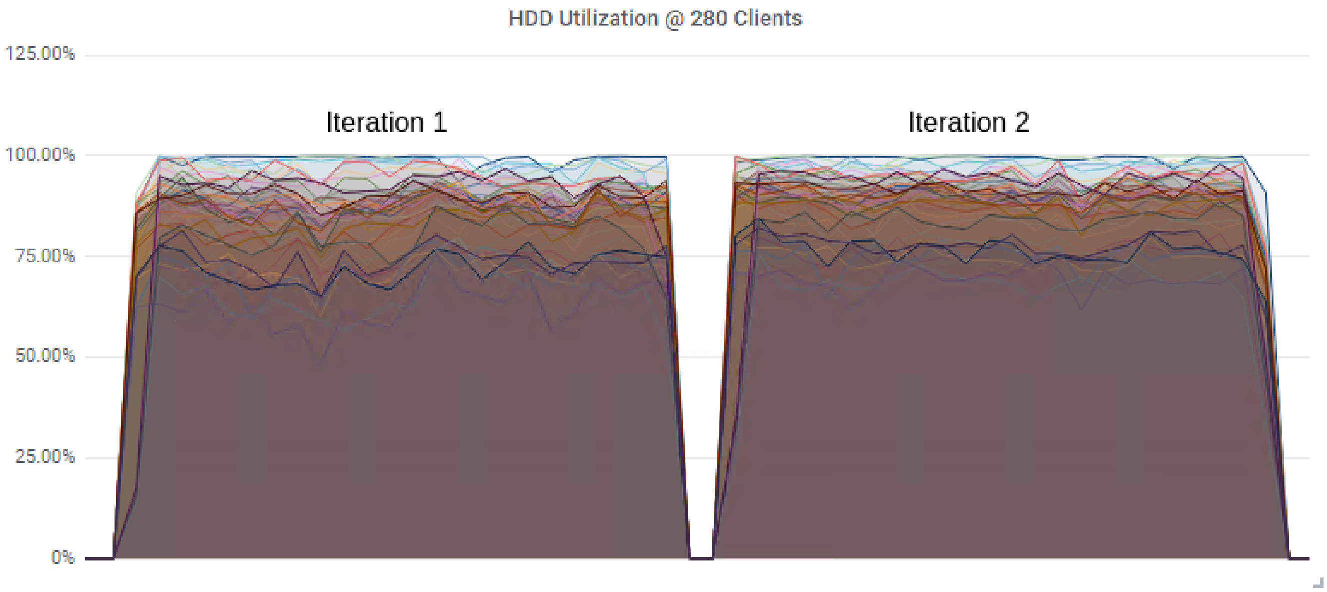


Figure 21: 1MB Object HDD Utilization

4MB read

The 4MB object size causes the system to reach throughput saturation very early on in the scaling process. Notice in the graph below how throughput saturates at only 80 clients. This is understandable since it is a much larger object size. The bottleneck is once again in the hard drives. All other system resources were not being utilized as much as that of HDDs.

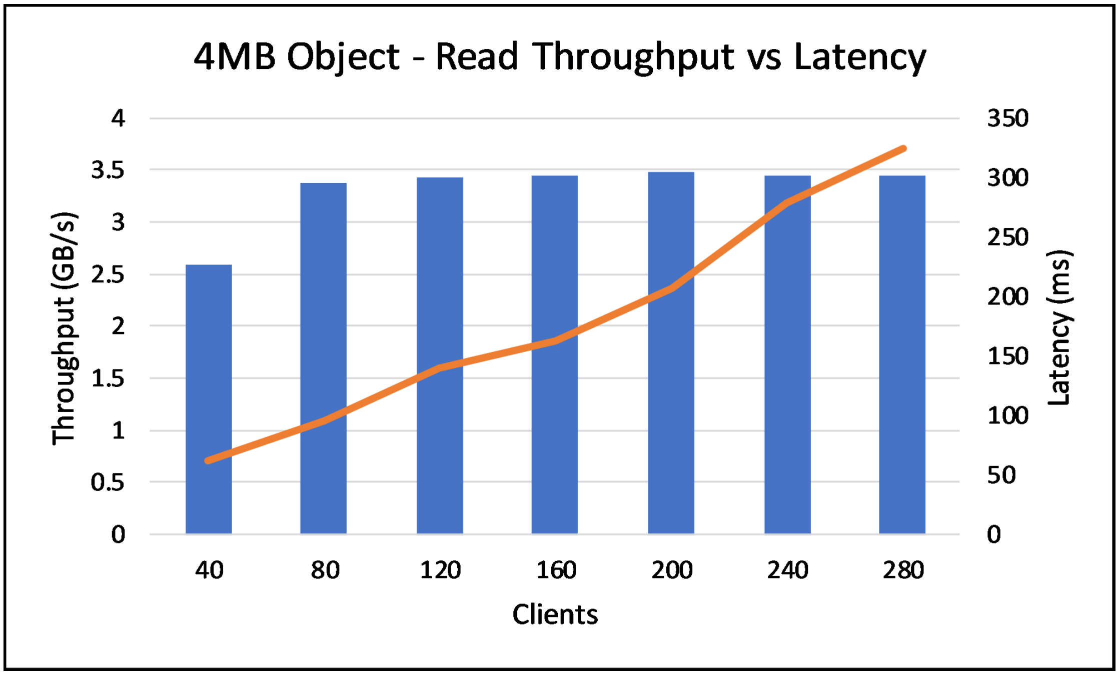


Figure 22: 4MB Object Read Throughput

We have described previously how read operations are directly serviced from the HDDs. Also, we know that for an object size as large as 4MB, it is expected to see a reduced I/O time utilization and more bandwidth utilization.

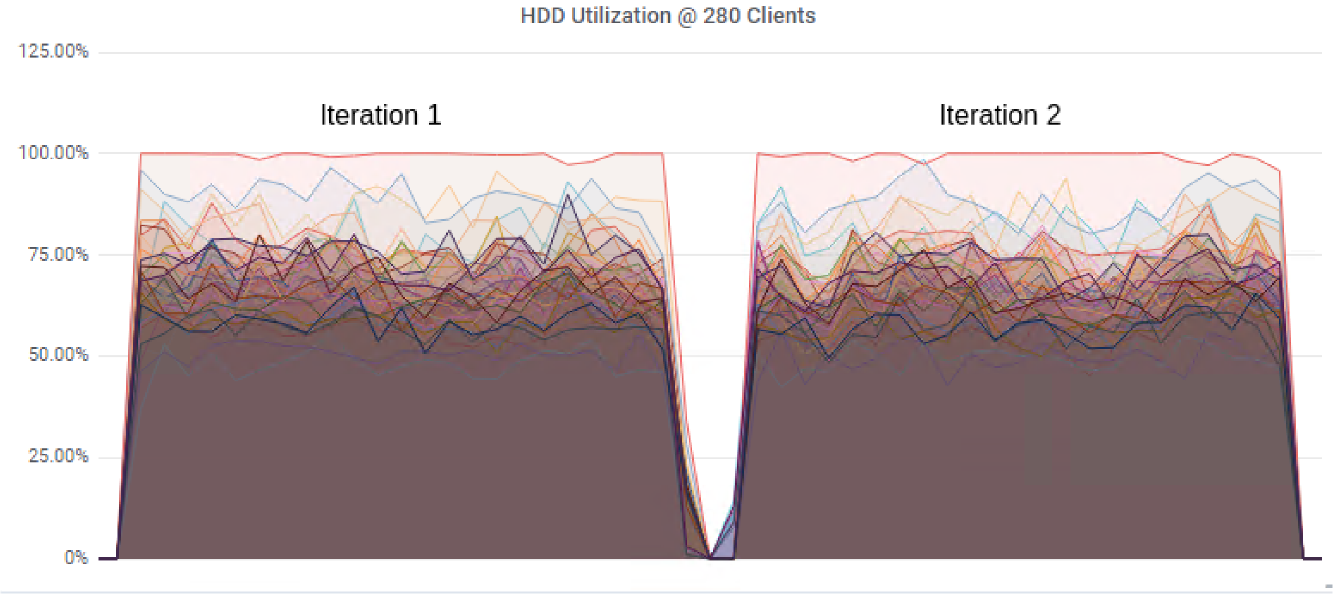


Figure 23: 4MB Object HDD Utilization

The graph below shows bandwidth utilization and the threshold has been marked at 90MB/s which is treated as the point of saturation of drives. This threshold is determined from the fact that due to large seek times, the maximum possible threshold of drives (as stated in hardware benchmark chapter) is not possible to be achieved. There is no OSD cache to facilitate the drive under read workloads, and hence they underperform their ideal numbers. This is again an expected outcome.

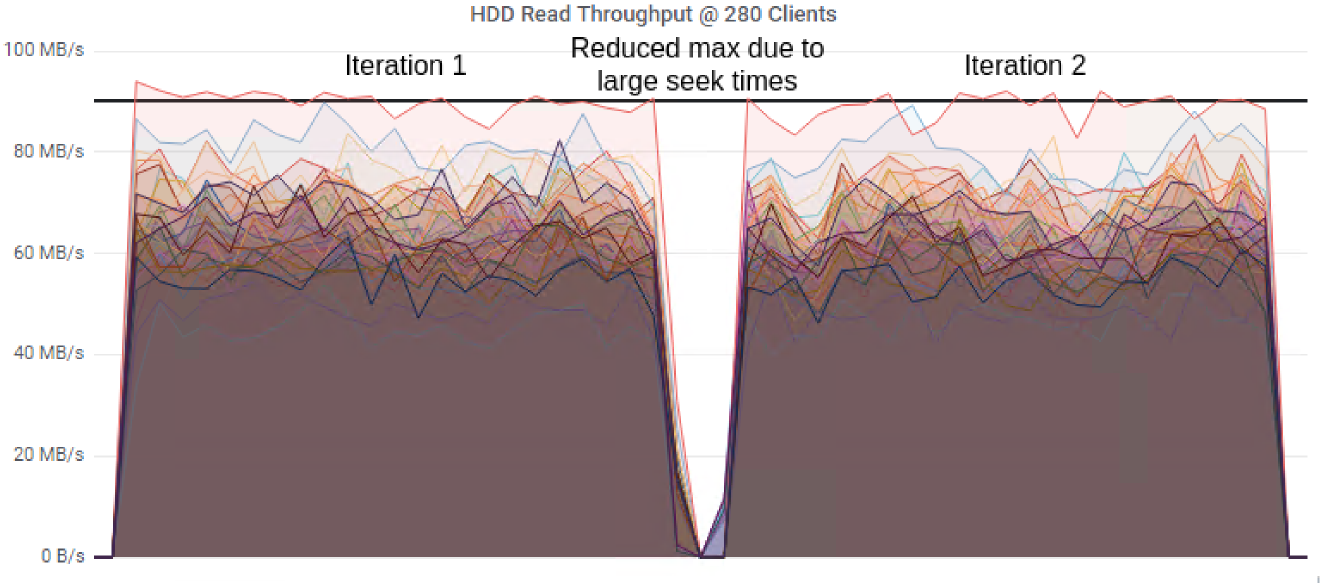


Figure 24: 4MB Object HDD Read Throughput

Also, we performed some tests prior to prefilling the cluster, and we observed the drive read throughputs rising as high as 120MB/s. This confirms our theory that larger seek times are detrimental to performance of throughput oriented workloads.

4MB write

The throughput saturates fully at only 120 clients, and even at 80 clients, is very close to saturation. The system resources when observed, once again point toward the HDDs being the bottleneck. The drive metrics are shown below.

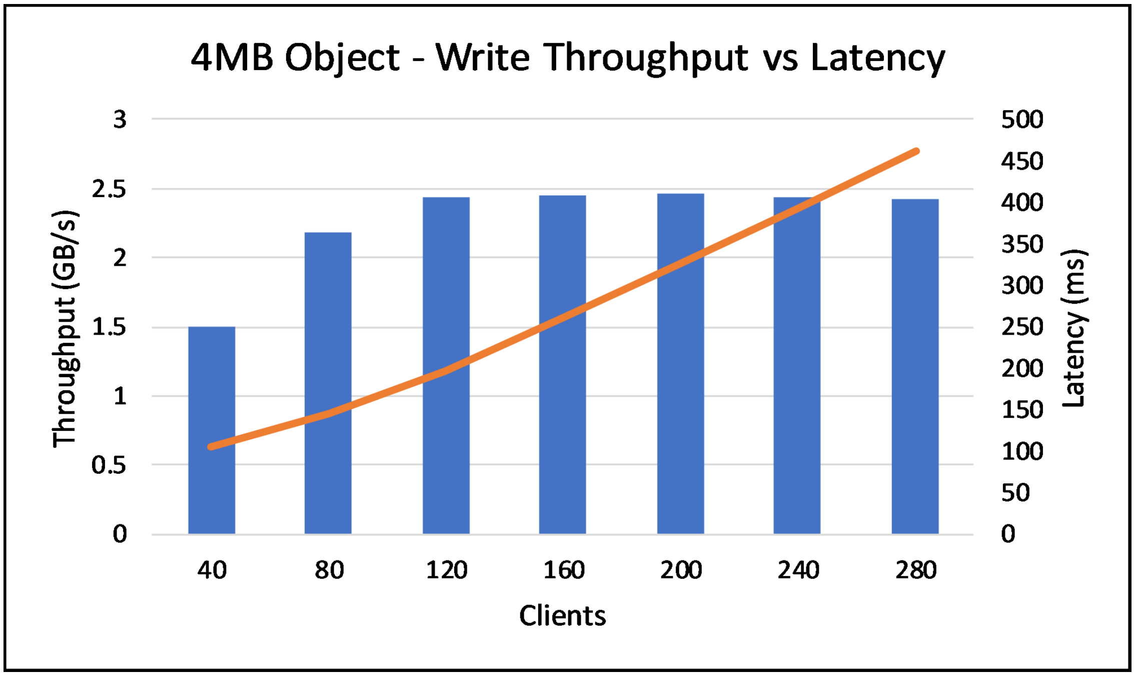


Figure 25: 4MB Object Write Throughput

In case of a write workload, since the WAL and OSD cache significantly increase the system throughput, the bottleneck shifts towards I/O times of HDDs. The graph below indicates how a large number of drives are saturated in their I/O times.

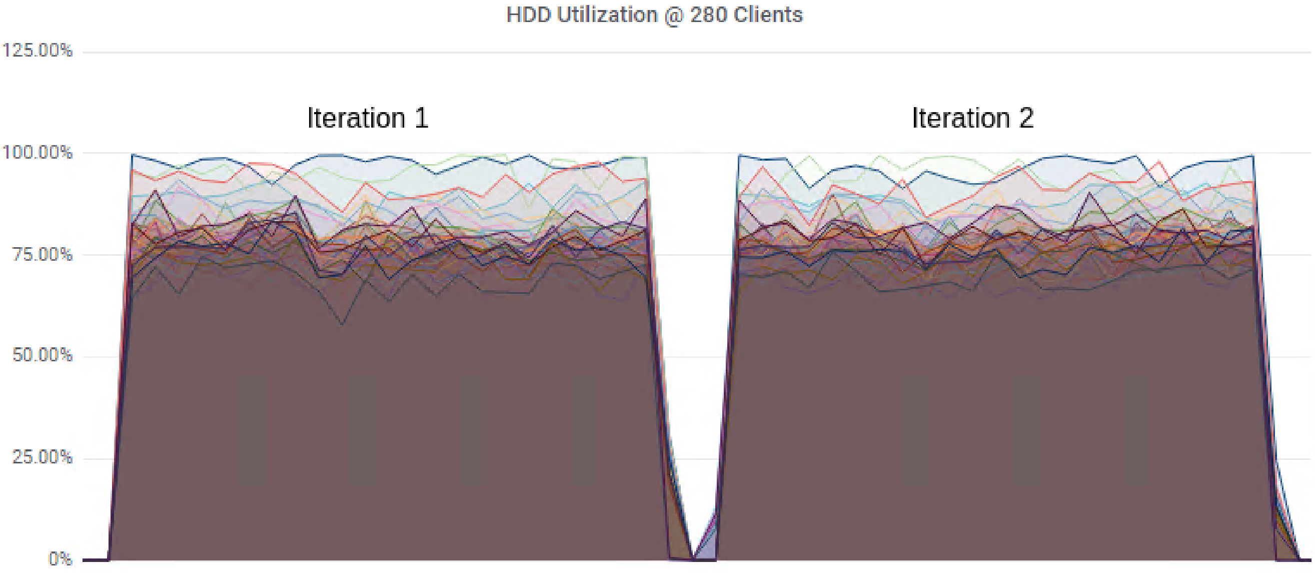


Figure 26: 4MB Object HDD Utilization

In addition, notice below that the throughput saturation has also been achieved. In essence, the hardware utilization has increased because of a workload that can not only greatly utilize the disk I/O time, but also the disk throughput.

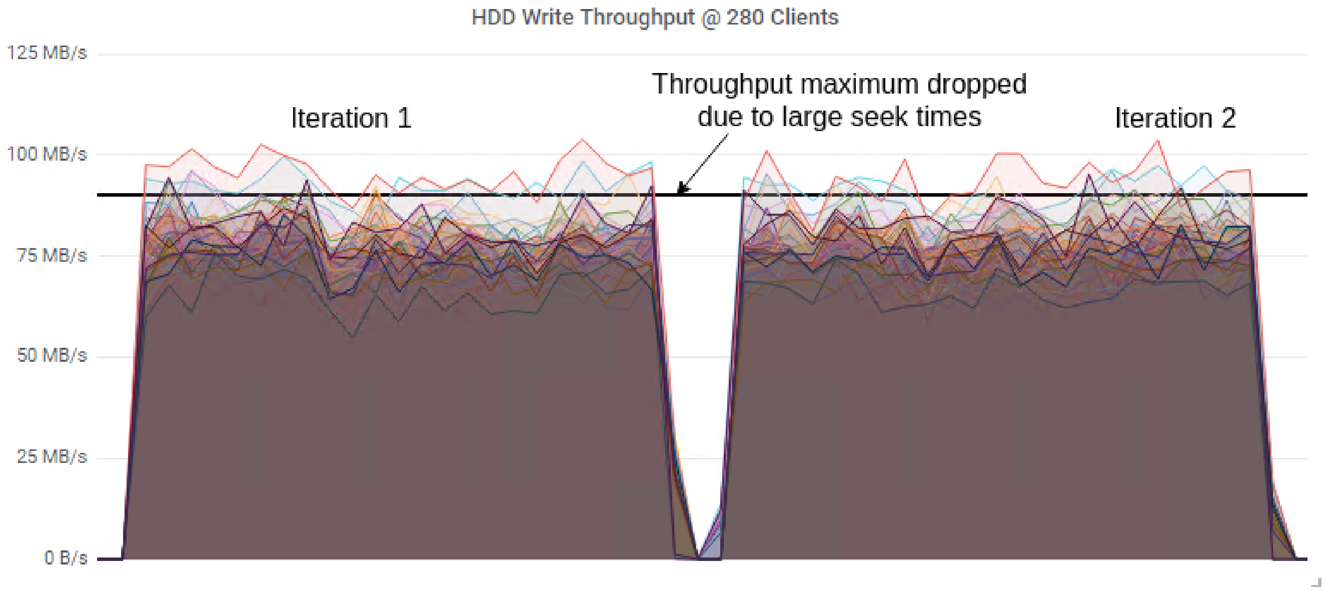


Figure 27: 4MB Object HDD Write Throughput

Analysis summary

The testing methodology exercised in the benchmarking process uses controlled, synthetic workloads that can be very different from those in production. However, it has been designed carefully to ensure that the cluster is independently analyzed from different perspectives. This allows for better insight into its properties. We can then use these independent properties to speculate onto cluster performance under custom workloads.

For instance, when we discuss 100% read and 100% write workloads, we can see how the cluster behaves given that all operations are of a single type for certain object sizes. Therefore, if we were to perform tests on object sizes close to these, we can use the hypotheses from the similar object size workloads, to explain the results of any relative workload. This not only makes them testing methodology more robust and versatile, but also enables designers to predict the cluster performance with a customized workload, which is more representative of a production environment.

The bottleneck was consistently observed to be the SATA HDDs. This is expected, since the 1:6 ratio of NVMe:HDD isn't enough to saturate the NVMe drives and an aggregate network capacity of 25GB/s (see note) is quite adequate for much larger node density clusters (more drives per node).

Note: 25GB/s is derived from:

- 2x 25GbE = 50GbE
- 50Gbits / 8 = 6.25GB
- 4x (storage nodes) 6.25GB = 25GB/s

One important thing worth highlighting is the relationship between drive utilization and drive bandwidth. Drive utilization is the percentage of time the drive was actively servicing requests. In the case of workloads of smaller object sizes, we generally observe a very high drive utilization as well as increased CPU usage (which is a consequence of that). This is because rapidly occurring small transaction magnify the effect of seek times causing the utilization to go high.

In contrast, the drive bandwidth corresponds to how much data the drive can process at a given time. This is generally a matter of concern in operations that comprise larger transactions; say 1MB or larger.

Also, since the drive isn't actively servicing operations, but rather accumulating them (as permitted by bandwidth), there's a lower utilization of drive compared with smaller object operations, and much higher bandwidth usage. This is why we tend to look at drive utilization for smaller object workloads, and drive bandwidth usage for larger object workloads during bottleneck analysis.

Table 29: Workload bottlenecks

Workload	Bottleneck	Suggested remedy
64KB read	SATA HDD drives	Faster drives or more storage nodes
64KB write		
1MB read		
1MB write		
4MB read		
4MB write		

Table 30: Bottleneck exposure by component

Area	Component	Bottleneck exposure	Comments
Ceph storage system	10TB SATA HDDs	Very likely (probable)	Disks were our bottlenecks in every set of tests
	HBA330 disk controller	None	Benchmark maximum throughputs stayed well below controller throughput limits
	Intel® Optane® P4800X (WAL,metadata)	Extremely low	• Benchmark tests never stressed Intel® P4600 NVMe to limits • P4800X has significantly higher write capabilities
Storage node motherboard	Intel® Xeon® Silver 4114 CPU	Possible	• Testing configuration (Xeon® Gold 6130, 16C) had significant extra CPU capacity • RA is more balanced after adjustment from tested CPU
	Memory	None	• Memory configuration has extra 40% capacity over total recommended • No memory pressure observed in any tests
Network	Ceph client 50GbE network	None	Benchmark tests never exceeded 70% of capacity
	Ceph cluster (replication) 50GbE network	None	Benchmark tests never exceeded 12% of capacity

Chapter 8

Conclusions

Topics:

This chapter presents the key takeaways of the study. It summarizes the results achieved and also reiterates the objectives of the work. We also discuss key traits of the Ready Architecture presented in the document.

Intel® Optane® P4800X guidance

Our architecture specifies the use of Intel® Optane® P4800X drives. However, our benchmark testing was performed with Intel® P4600 NVMe drives. This was because Intel® Optane® P4800X drives were not available for order at the time our servers were acquired. The following table provides published performance ratings for both drives.

Table 31: Performance ratings of Intel® P4600 and Intel® Optane P4800X

Metric	Intel® P4600 2TB AIC	Intel® Optane® P4800X 750GB AIC	Change
Sequential read (up to)	3200 MB/s	2500 MB/s	-22%
Sequential write (up to)	1575 MB/s	2200 MB/s	+40%
Random read	610,000 IOPS	550,000 IOPS	-10%
Random write	196,650 IOPS	550,000 IOPS	+180%
Latency read	85 microseconds	10 microseconds	-88%
Latency write	15 microseconds	10 microseconds	-33%

As shown in the table above, the Intel® Optane® 4800X delivers dramatically higher throughput for random write workloads. Since this device is used for Ceph metadata (WAL and RocksDB), we would expect significantly higher write throughputs, particularly for workloads involving smaller block sizes.

Conclusions

The performance- and capacity-optimized Ready Architecture presented in this document is well suited for use cases where a blend of high performance, high capacity, and cost-effectiveness are critical design factors. With the high resource density of Dell R740xd servers, the colocation of Ceph services is a very attractive choice for RA design. The combination of RHCS 3.2, RHEL 7.6, and 50GbE (dual bonded 25GbE) networking provides a solid foundation for a performance- and capacity-optimized Ceph cluster. Additionally, the use of containerized Ceph daemons worked well in this study.

Even with only four storage nodes, we were able to achieve the following performance results:

Table 32: Performance highlights

Workload	Result
64KB 100% read	252 MB/s
64KB 100% write	343 MB/s
1MB 100% read	2.7 GB/s
1MB 100% write	1.9 GB/s
4MB 100% read	3.47 GB/s
4MB 100% write	2.46 GB/s

The testing methodology adopted for benchmarking of the architecture has been developed rigorously. This is to ensure that it is generic in nature and is easy to exercise in customized environments. Also, the choice of workloads is based on well-known community standards. This also assists the process of performance comparison with other architectures.

The architecture is very flexible, allowing it to be tweaked to meet various capacity/price objectives. This will then involve a choice of a high-speed caching device and a slower data device as desired. This is possible since the R740xd can support various hardware configurations. Also, the testing methodology and architecture design process can be replicated for other devices as well. This is what we mean by a generic design. It must be noted that changes made can introduce new performance bottlenecks (or shift them from one place to another), so care must be exercised.

The workload specifications can also affect the choice of other components of the architecture. For instance, with workloads that are comprised primarily of large object sizes (say 4MB or higher), the drive throughput might be the primary suspect in performance degradation. But for other, less bandwidth hungry workloads, (say 64KB and smaller object sizes) the drive rotational speed might limit performance and in any case the presented 50GbE (dual bonded 25GbE) network design should be appropriate.

The objective of the work was to provide a detailed insight into the capabilities of the state-of-the-art hardware as well as RHCS 3.2 software, provide a robust and generic methodology of testing, and finally, point out critical design parameters. Most importantly, we present a flexible architecture that can be easily adapted to the appropriate capacity/price targets.

Appendix A

References

Topics:

Note: If you need additional services or implementation help, please call your Dell EMC sales representative.

Bill of Materials (BOM)

Table 33: Bill of Materials (BOM) - R740xd Storage Nodes

Component	Configuration
Server model	PowerEdge R740xd Server
BIOS	Performance Optimized
Remote admin access	iDRAC9 Enterprise
Motherboard risers	Riser Config 6, 5 x8, 3 x16 PCIe slots
Chassis	Chassis with up to 12 x 3.5” hard drives for 2CPU configuration
CPU	Intel® Xeon® Silver (see note) 4114 2.2G,10C/20T,10.4GT/s,14M Cache,Turbo,DDR4-2666
RAM	192GB (12x 16GB RDIMM), 2666MT/s, dual rank
Disk controller	HBA330 controller, 12Gbps adapter, low profile
Data drives	12x 10TB (see caution) 7.2K RPM SATA 6Gbps 512e 3.5in Hot-plug Drive
1GbE NIC	I350 QP 1Gb Ethernet, Network Daughter Card
25GbE NICs	2x Intel® XXV710 Dual Port 25GbE SFP28 PCIe Adapter, Full Height
System storage (OS)	BOSS controller card + with 2 M.2 Sticks 240G (RAID 1),FH
High-speed storage devices	2x Intel® Optane® (see note) P4800X 750GB AIC

Note: Our performance testing was conducted with Gold 6130 CPUs because we anticipated experimentation with configurations that require greater CPU resources. Please use Silver 4114 instead of Gold 6130.

CAUTION: Please consult with Red Hat to discuss support for use of drives with capacity above 8TB.

Note: Our performance testing was conducted with P4600 NVMe devices because the P4800X was not orderable at the time the servers were acquired. Please use P4800X instead of P4600.

Table 34: Bill of Materials (BOM) - R640 Admin Node

Component	Configuration
Server	PowerEdge R640 Server
Remote admin access	iDRAC9 Enterprise
Storage drives	8x 600GB 10K RPM SAS 12Gbps 512n 2.5in Hot-plug Hard Drive
Chassis	2.5” Chassis with up to 8 Hard Drives and 3PCIe slots
BIOS	Performance Optimized
RAM	192GB (12x 16GB RDIMM), 2933MT/s, Dual Rank

Component	Configuration
Disk controller	PERC H740P RAID Controller, 8GB NV Cache, Minicard
Motherboard risers	Riser Config 2, 3 x16 LP
CPU	2x Intel® Xeon® Gold 6226 2.7G,12C/24T, 19M Cache,Turbo,HT (125W) DDR4-2933
25GbE NICs	1x Intel® XXV710 Dual Port 25GbE SFP28 PCIe Adapter, Low Profile
1GbE NIC	I350 QP 1Gb Ethernet, Network Daughter Card

Tested BIOS and firmware

⚠ **CAUTION:** Ensure that the firmware on all servers and switches are up to date. Otherwise, unexpected results may occur.

Table 35: Tested server BIOS and firmware versions

Product	Version
BIOS	1.4.9
iDRAC with Lifecycle controller	3.21.23.22
Intel® XXV710 NIC	18.5.17
PERC H740P (R640)	05.3.3-1512
BOSS-S1 (R740xd)	2.3.13.1084

Table 36: Tested switch firmware versions

Product	Version
S3048-ON firmware	Dell OS 9.9(0.0)
S5248F-ON firmware	Cumulus 3.7.1

Configuration details

Configuration details for object storage cluster

Linux network bonding

```
mode=802.3ad miimon=100 xmit_hash_policy=layer3+4 lacp_rate=1
```

all.yml

```
fetch_directory: /root/ceph-ansible-keys
cluster: ceph
mon_group_name: mons
osd_group_name: osds
rgw_group_name: rgws
mgr_group_name: mgrs
check_firewall: False
```

```
redhat_package_dependencies:
  - python-pycurl
  - hdparm
  - python-setuptools
ntp_service_enabled: true
ceph_origin: repository
ceph_repository: rhcs
ceph_rhcs_version: 3
ceph_repository_type: cdn

ceph_docker_image: rhceph/rhceph-3-rhel7
containerized_deployment: true
ceph_docker_registry: registry.access.redhat.com

fsid: "{{ cluster_uuid.stdout }}"
generate_fsid: true
monitor_address_block: 192.168.170.0/24
ip_version: ipv4
mon_use_fqdn: false
public_network: 192.168.170.0/24
cluster_network: 192.168.180.0/24
osd_mkfs_options: -f -i size=2048
osd_mount_options_xfs: noatime, largeio, inode64, swalloc
osd_objectstore: bluestore
osd_memory_target: 8589934592
cephx: false

radosgw_frontend_type: civetweb # For additional frontends see: http://
docs.ceph.com/docs/mimic/radosgw/frontends/
radosgw_civetweb_port: 7480
radosgw_frontend_port: "{{ radosgw_civetweb_port if radosgw_frontend_type ==
 'civetweb' else '8080' }}"
radosgw_address_block: 192.168.170.0/24

ceph_conf_overrides:
  global:
    auth_cluster_required: none
    auth_service_required: none
    auth_client_required: none
    mon_allow_pool_delete: true
    osd_pool_default_size: 2
  osd:
    bluestore_block_db_size: 268435456000
    bluestore block wal size: 2147483648

ceph_tcmalloc_max_total_thread_cache: 134217728
```

osds.yml

```
ceph_osd_docker_cpu_limit: 3
osd_auto_discovery: false
osd_scenario: lvm
osd_objectstore: bluestore
lvm_volumes:
  - data: data-lva
    data_vg: vg_sda
    db: db-lva
    db_vg: vg_nvme0n1
    wal: wal-lva
    wal_vg: vg_nvme0n1
  - data: data-lvb
    data_vg: vg_sdb
  db: db-lvb
  db_vg: vg_nvme0n1
  wal: wal-lvb
  wal_vg: vg_nvme0n1
- data: data-lvc
  data_vg: vg_sdc
  db: db-lvc
  db_vg: vg_nvme0n1
  wal: wal-lvc
  wal_vg: vg_nvme0n1
- data: data-lvd
  data_vg: vg_sdd
  db: db-lvd
  db_vg: vg_nvme0n1
  wal: wal-lvd
  wal_vg: vg_nvme0n1
- data: data-lve
  data_vg: vg_sde
  db: db-lve
  db_vg: vg_nvme0n1
  wal: wal-lve
  wal_vg: vg_nvme0n1
- data: data-lvf
  data_vg: vg_sdf
  db: db-lvf
  db_vg: vg_nvme0n1
  wal: wal-lvf
  wal_vg: vg_nvme0n1
- data: data-lvg
  data_vg: vg_sdg
  db: db-lvg
  db_vg: vg_nvme1n1
  wal: wal-lvg
  wal_vg: vg_nvme1n1
- data: data-lvh
  data_vg: vg_sdh
  db: db-lvh
  db_vg: vg_nvme1n1
  wal: wal-lvh
  wal_vg: vg_nvme1n1
- data: data-lvi
  data_vg: vg_sdi
  db: db-lvi
  db_vg: vg_nvme1n1
  wal: wal-lvi
  wal_vg: vg_nvme1n1
- data: data-lvj
  data_vg: vg_sdj
  db: db-lvj
  db_vg: vg_nvme1n1
  wal: wal-lvj
  wal_vg: vg_nvme1n1
- data: data-lvk
  data_vg: vg_sdk
  db: db-lvk
  db_vg: vg_nvme1n1
  wal: wal-lvk
  wal_vg: vg_nvme1n1
- data: data-lvl
  data_vg: vg_sdl
  db: db-lvl
  db_vg: vg_nvme1n1
  wal: wal-lvl
  wal_vg: vg_nvme1n1
  - data: data-lv0
    data_vg: vg_nvme0n1
    db: db-lv0
    db_vg: vg_nvme0n1
    wal: wal-lv0
    wal_vg: vg_nvme0n1
  - data: data-lv1
    data_vg: vg_nvme1n1
    db: db-lv1
    db_vg: vg_nvme1n1
    wal: wal-lv1
    wal_vg: vg_nvme1n1
```

rgws.yml

```
rgw_thread_pool_size: 100
ceph_rgw_docker_cpu_limit: 18
```

mgrs.yml

```
ceph_mgr_modules: [dashboard, prometheus]
ceph_mgr_docker_cpu_limit: 2
```

mons.yml

```
mon_group_name: mons
ceph_mon_docker_cpu_limit: 2
```

Benchmark details

Dropping caches

```
For BlueStore, you need to restart every OSD container to clear OSD cache

Set the noout flag on the cluster
# ceph osd set noout

The following script can be modified as needed

for server in ostor0 ostor1 ostor2 ostor3; do
ssh root@${server} "for osd in \$(docker ps | awk '{print \$10}' | grep
 osd);do docker container restart \$osd; done"
done

Clear the noout flag when done
# ceph osd unset noout
```

To learn more

For more information on Dell EMC Service Provider Solutions, visit *https://www.dellemc.com/en-us/service-providers/index.htm*

Copyright © 2019 Dell EMC or its subsidiaries. All rights reserved. Trademarks and trade names may be used in this document to refer to either the entities claiming the marks and names or their products. Specifications are correct at date of publication but are subject to availability or change without notice at any time. Dell EMC and its affiliates cannot be responsible for errors or omissions in typography or photography. Dell EMC's Terms and Conditions of Sales and Service apply and are available on request. Dell EMC service offerings do not affect consumer's statutory rights.

Dell EMC, the DELL EMC logo, the DELL EMC badge, and PowerEdge are trademarks of Dell EMC.

Glossary

Ansible

Ansible is an open source software utility used to automate the configuration of servers.

API

Application Programming Interface is a specification that defines how software components can interact.

BlueStore

BlueStore is a new OSD storage backend that does not use a filesystem. Instead, it uses raw volumes and provides for more efficient storage access.

BMC/iDRAC Enterprise

Baseboard Management Controller. An on-board microcontroller that monitors the system for critical events by communicating with various sensors on the system board, and sends alerts and log events when certain parameters exceed their preset thresholds.

BOSS

The Boot Optimized Storage Solution (BOSS) enables customers to segregate operating system and data on Directly Attached Storage (DAS). This is helpful in the Hyper-Converged Infrastructure (HCI) and Software-Defined Storage (SDS) arenas, to separate operating system drives from data drives, and implement hardware RAID mirroring (RAID1) for OS drives.

Bucket data

In the context of RADOS Gateway, this is the storage pool where object data is stored.

Bucket index

In the context of RADOS Gateway, this is the storage pool that houses metadata for object buckets.

CBT

Ceph Benchmarking Tool

Cluster

A set of servers that can be attached to multiple distribution switches.

COSBench

An open source tool for benchmarking object storage systems.

CRC

Cyclic redundancy check. This is a mechanism used to detect errors in data transmission.

CRUSH

Controlled Replication Under Scalable Hashing. This is the name given to the algorithm used by Ceph to maintain the placement of data objects within the cluster.

Daemon

Daemon is a long-running Linux process that provides a service.

DIMM

Dual In-line Memory Module

FileStore

FileStore is the original OSD storage backend that makes use of the XFS filesystem.

FIO

Flexible IO Tester (synthetic load generation utility)

Grafana

Grafana is open-source software that provides flexible dashboards for metrics analysis and visualization.

iPerf

iPerf is an open-source tool that is widely used for network performance measurement.

JBOD

Just a Bunch of Disks

LAG

Link Aggregation Group

LINPACK

LINPACK is a collection of benchmarks used to measure a system's floating point performance.

MON

MON is shorthand for the Ceph Monitor daemon. This daemon's primary responsibility is to provide a consistent CRUSH map for the cluster.

MTU

Maximum Transmission Unit

NFS

The Network File System (NFS) is a distributed filesystem that allows a computer user to access, manipulate, and store files on a remote computer, as though they resided on a local file directory.

NIC

Network Interface Card

Node

One of the servers in the cluster

NUMA

Non-Uniform Memory Access

NVMe

Non-Volatile Memory Express is a high-speed storage protocol that uses PCIe bus

OSD

Object Storage Daemon is a daemon that runs on a Ceph storage node and is responsible for managing all storage to and from a single storage device (or a partition within a device).

PG

Placement Group is a storage space used internally by Ceph to store objects.

Prometheus

Prometheus is an open-source software that provides metrics collection into a time-series database for subsequent analysis.

RACADM

Remote Access Controller ADMinistration is a CLI utility that operates in multiple modes (local, SSH, remote desktop) to provide an interface that can perform inventory, configuration, update as well as health status check on Dell PowerEdge servers.

RADOS

RADOS is an acronym for Reliable Autonomic Distributed Object Store and is the central distributed storage mechanism within Ceph.

RADOSGW

RADOS Gateway provides S3 and Swift API compatibility for the Ceph cluster. Sometimes also written as RGW.

RBD

RADOS Block Device is a block device made available in Ceph environment using RADOS.

RGW

RADOS Gateway provides S3 and Swift API compatibility for Ceph cluster. Sometimes also written as RADOSGW.

RocksDB

RocksDB is an open source key-value database that is used internally by BlueStore backend to manage metadata.

S3

The public API provided by Amazon's S3 Object Storage Service.

SDS

Software-defined storage (SDS) is an approach to computer data storage in which software is used to manage policy-based provisioning and management of data storage, independent of the underlying hardware.

Storage Node

A server that stores data within a clustered storage system.

Swift

The public API provided by OpenStack Swift object storage project.

U

U used in the definition of the size of the server, example 1U or 2U. A "U" is a unit of measure equal to 1.75 inches in height. This is also often referred to as a rack unit.

WAL

WAL is an acronym for the write-ahead log. The write-ahead log is the journaling mechanism used by the BlueStore backend.