DELL EMC

DELL EMC® DSS 7000 PERFORMANCE & SIZING GUIDE FOR RED HAT® CEPH STORAGE 2

Optimized software-defined storage for demanding workloads

ABSTRACT

This technical white paper provides an overview of the Dell EMC DSS 7000 server performance results with Red Hat Ceph Storage. It covers the advantages of using Red Hat Ceph Storage on Dell EMC servers to provide high scalability, enhanced ROI cost benefits, and support for unstructured data. This paper also provides specific hardware configuration recommendations for the DSS 7000 when running Red Hat Ceph Storage workloads in a variety of environments.

November, 2016

TABLE OF CONTENTS

REVISIONS

Date	Description
November 2016	Initial release

EXECUTIVE SUMMARY

Today's data storage requirements are staggering and are growing at an ever-accelerating rate. These demanding capacity and growth trends are fueled in part by the enormous expansion in unstructured data, including music, image, video, and other media; database backups, log files, and other archives; financial and medical data; and large data sets often termed "Big Data". These burgeoning data storage requirements are expected to increase exponentially with the continued rise of the Internet of Things (IoT) yet customer expectations for highly-reliable, high performance solutions are greater than ever. As IT organizations struggle with how to manage petabytes and even exabytes of ever-growing digital information, the adoption of cloud-like storage models is becoming more common in modern data centers. One answer is the software known as Ceph.

Ceph is an open source distributed object storage system designed to provide high performance, reliability, and massive scalability. Ceph implements object storage on a distributed cluster and provides interfaces for object-, block- and file-level storage. Ceph provides for completely distributed operation without a single point of failure as well as scalability to the multi-petabyte level. Ceph replicates data to guard against the disk failures inherent in today's cloud infrastructures. As a result of its design, the system is both self-healing and self-managing thus minimizing administration time and related costs. Since Ceph uses general-purpose standard server hardware controlled by software whose features are exposed through application programming interfaces (APIs), it is considered to be a type of software-defined storage (SDS).

Red Hat Ceph Storage is an enterprise-ready implementation of Ceph that provides a single platform solution for software-defined storage that is open, adaptable, massively scalable, technologically advanced, and supported worldwide. Red Hat Ceph Storage combines innovation from the open source community with the backing of Red Hat engineering, consulting, and support. It includes tight integration with OpenStack services and was built from the ground up to deliver next-generation storage for cloud and emerging workloads.

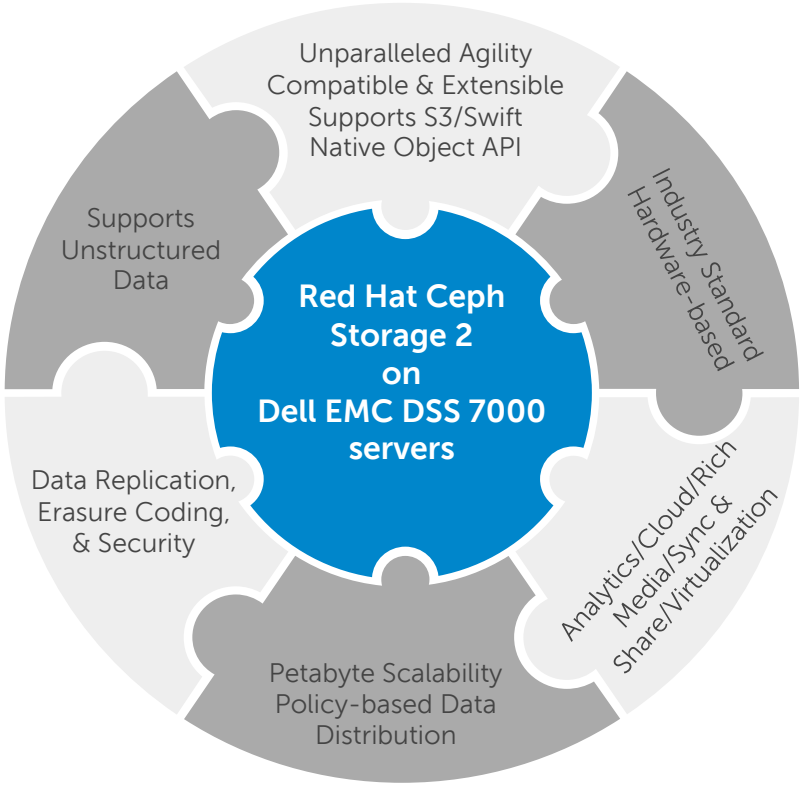
This technical white paper provides performance and sizing guidelines for Red Hat Ceph Storage running on Dell EMC servers, specifically the Dell EMC DSS 7000, based on extensive testing performed by Dell EMC engineering teams. The DSS 7000 is a cost-optimized, scale-out storage server platform that provides high capacity and scalability along with an optimal balance of storage utilization, performance, and cost.

AUDIENCE

This paper is intended for storage architects, engineers, and IT administrators who want to explore the advantages of using Red Hat Ceph Storage on Dell EMC servers and who need to design and plan implementations using proven best practices.

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.



Red Hat Ceph Storage makes use of industry-standard servers that form Ceph storage 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.

- Replicated storage pools make full copies of stored objects which is ideal for quick recovery. In a replicated storage pool, the Ceph configuration defaults to a replication factor of three, i.e. three copies of the data exist on three separate Ceph nodes.
- Erasure-coded storage pools provide a single copy of data plus parity which is useful for archival 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 its CRUSH algorithm after the erasure coding process.

For more information on designing scalable workload-optimized Ceph clusters, please see the configuration guide at https://access.redhat.com/documentation/en/red-hat-ceph-storage/2/paged/configuration-guide/

The Dell EMC DSS 7000 is a highly-versatile, ultra-dense storage server built to provide best-in-class cost efficiency for object and block storage for scale-out datacenters, cloud builders, file storage and archival. Featuring the latest Intel® Xeon® E5-2600 v4 processors, the DSS 7000 server provides cloud builders the performance they need and the manageability they demand.

	Dell EMC DSS 7000 dense storage server
Form factor	4U rack
Nodes per chassis	2
Processor	Intel® Xeon® processor E5-2600 v4 product family
Processor sockets Per Node	2
~~**Dimensions**~~	~~H: 173.8 mm (6.84 in) x W: 434 mm (17.00 in) x L: 1243.3 mm (48.94 in)~~
Cache	2.5 MB per core; core options: 8, 12, 14, 16, 18
Chipset	Intel C610 series chipset
Memory Per Node	Up to 384 GB (12 DIMM slots): 16 GB/32 GB DDR4 up to 2400MT/s
I/O slots Per Node	Up to 4 x PCIe 3.0 slots One dedicated to HBA/RAID controller for data drives
RAID controllers	Internal controllers: Broadcom MegaRAID SAS 9361-8i, Broadcom 9311-8i, MicroSemi 8805
Drive bays Per Node	Internal hard drive bay and hot-plug backplane: Up to 45 x 3.5" SATA drives per node (90 drives per chassis). Up to 6 800G or 1.6TB SSD per chassis Up to 2 x 2.5" SATA drives (Hot swap boot devices)
Maximum internal storage	SATA: Up to TB with 90 x 3.5" TB hot-plug SATA HDD
Embedded NIC	4 x 1GbE
Power supply unit (PSU)	Platinum efficiency 1100 W, and 1600 W AC PSU
Availability	ECC memory, hot-plug hard drives, hot-plug redundant cooling, hot-plug redundant power, spare rank, proactive systems management alerts, iDRAC8 with Lifecycle Controller
Remote management	iDRAC8 with Lifecycle Controller, iDRAC8 Express
Rack support	Sliding rails for tool-less mounting in 4-post racks with square or unthreaded round holes or tooled mounting in 4-post threaded hole racks, with support for optional tool-less cable management arm.
Recommended support	Dell EMC ProSupport Plus for critical systems or Dell EMC ProSupport for premium hardware and software support for your PowerEdge solution. Consulting and deployment offerings are also available. Contact your Dell EMC representative today for more information. Availability and terms of Dell EMC Services vary by region. For more information, visit **Dell.com/ServiceDescriptions**.
Dimensions	Height: 173.8mm Width: 434mm Depth – Chassis only: 1098.4mm Depth with rails and CMA: 1242.68mm (48.94 in) Weight (maximum configuration): 129.5 kg (285 lbs.) (with 90 x 3.5" HDDs and 2x server nodes) Weight (empty): 57.1 kg (125.88lbs.)

The DSS 7000 storage server offers a number of key advantages for running Red Hat Ceph Storage today:

Maximize storage density

Provide massive object- or block-level data storage with up to ninety hot-serviceable 3.5-inch SATA hard drives, each up to 10 TB in capacity, for nearly a petabyte of storage per 4U chassis. In addition, two 2.5-inch SATA boot drives provide additional capability for high-performance storage within each node for rapid access of critical data.

Discover greater operational reliability

Increase flexibility and performance for key workloads with support for up to four PCIe Gen 3.0 slots per node with dual 40Gbps Ethernet connectivity to eliminate costly bottlenecks. Two Platinum power supplies per node provide optimal power.

Increase versatility

This innovative multi-node design is based on a standard x86 platform with support for industry-leading operating systems and commercially available peripherals.

TESTING OVERVIEW

This technical white paper discusses the test results and configuration details for running Red Hat Ceph Storage on the DSS 7000 platform. The primary focus of the testing is related to capacity-optimized object storage specific to this platform.

Details of other Dell EMC solutions for Ceph are documented here:
http://en.community.dell.com/techcenter/cloud/m/dell_cloud_resources/20442913

The Ceph Benchmarking Toolkit (CBT) was used to generate and measure IO workload. Both read and write operations were performed to test throughput differences between replicated and erasure-coded methods.

The table below shows a summary of the DSS 7000 and Red Hat Ceph Storage configurations used for benchmark tests used for this paper. All testing was performed against Red Hat Ceph Storage.

Configuration	Brief description
DSS 7000 45+2, 3x Replication	DSS 7000 with 45 hard disk drives (HDDs) per node and two Intel P3700 PCIe add-in NVMe (SSDs), 3X data replication and single-drive RAID0 mode
DSS 7000 45+2, EC 4+2	DSS 7000 with 45 hard disk drives (HDDs) per node and two PCIe add-in NVMe (SSDs), erasure-coding and single-drive RAID0 mode

Note: Though the DSS 7000 server provides flexibility in the layout and configuration of I/O subsystems, the combinations described in the table above were selected on the basis of performance data of different configuration variations tested.

The benchmark results will enable the reader to compare server throughput based on replication versus erasure-coded modes.

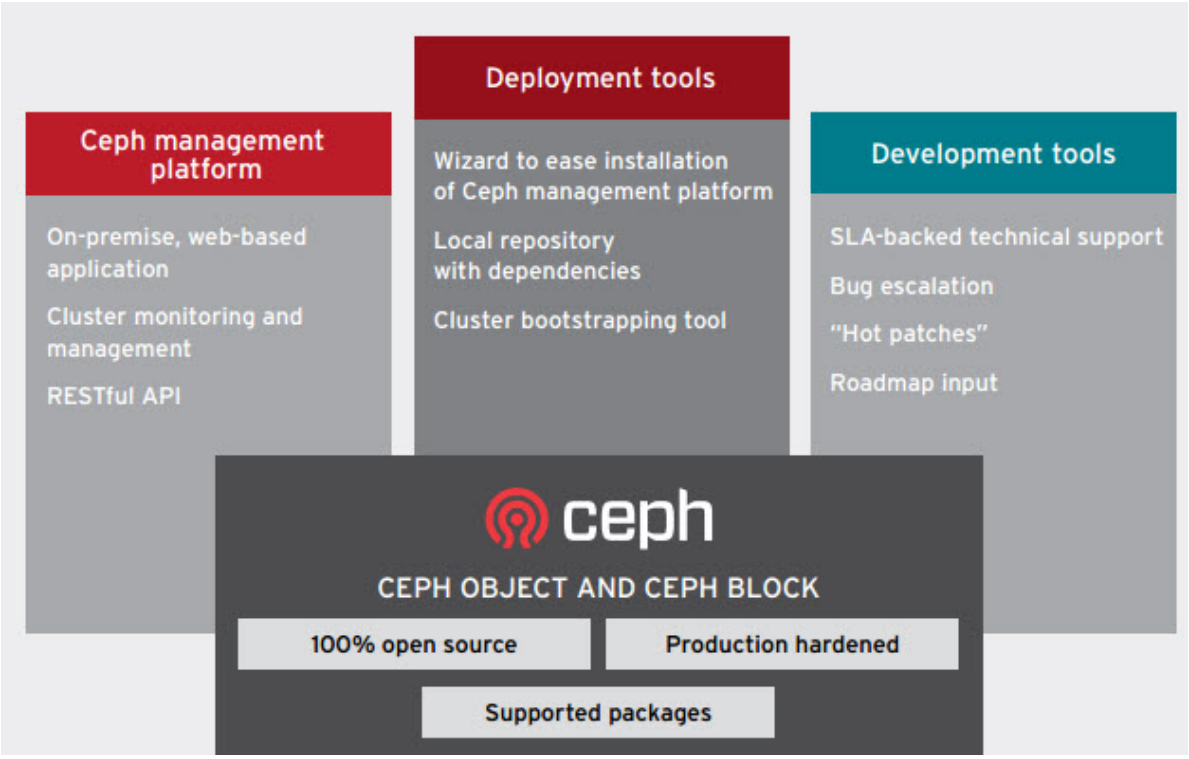
OVERVIEW OF RED HAT CEPH STORAGE

A Ceph storage cluster is built from large numbers 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

Red Hat Ceph Storage provides enterprise block and object storage access for archival, rich media, and cloud infrastructure workloads such as OpenStack. A few advantages of Red Hat Ceph Storage are shown below:

- 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 & Quality Assurance Programs
- Red Hat Knowledgebase (articles, tech briefs, videos, documentation), Automated Services



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, Red Hat Ceph Storage handles data management so businesses can focus on improving application availability. Some of the properties include:

- Scaling to petabytes or even exabytes of capacity
- No single point of failure in the cluster
- Lower capital expenses (CapEx) by running on general-purpose server hardware
- Lower operational expenses (OpEx) by self-managing and self-healing

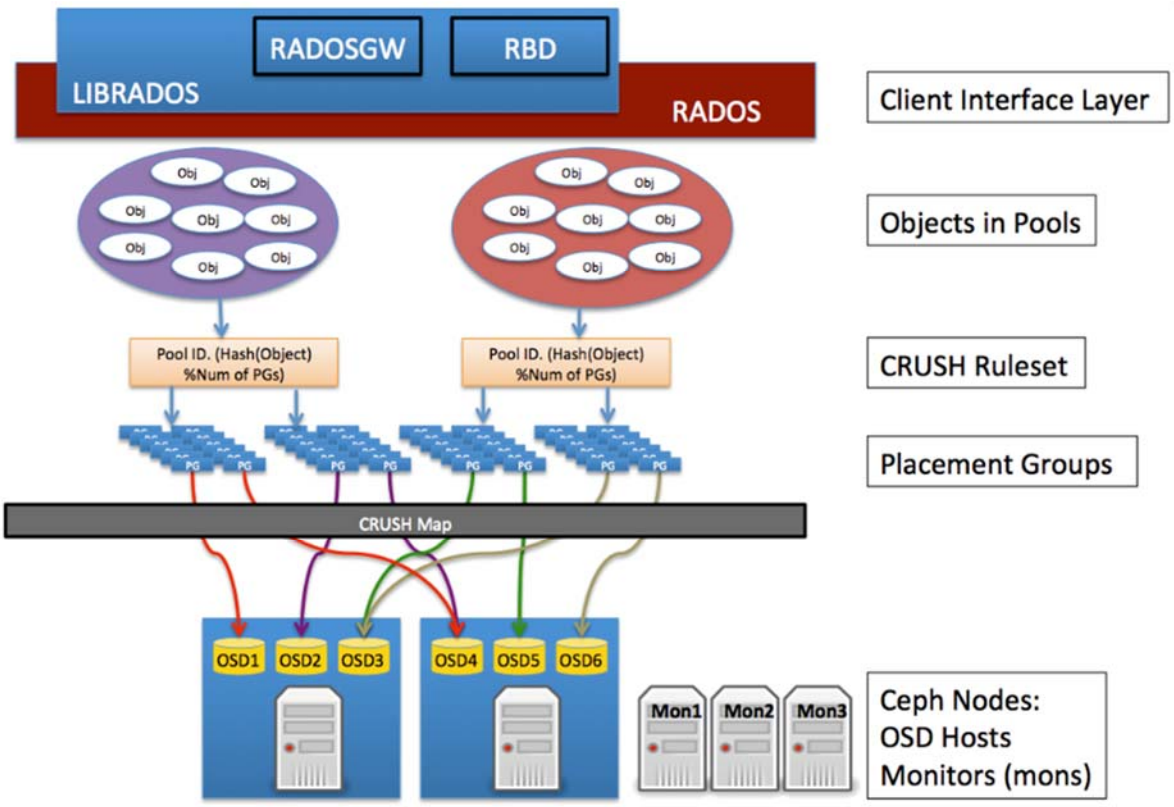
Red Hat Ceph Storage is a software-defined storage platform that supports multiple types of workloads in the same Ceph cluster. The underlying industry standard hardware in a Ceph cluster can be configured to handle high IOPS workloads, high throughput workloads and cost / capacity optimized workloads.

The table below provides a matrix of different Ceph cluster design factors, optimized by workload category

Optimization criteria	Potential attributes	Example uses
Capacity-optimized	• Lowest cost per TB • Lowest BTU per TB • Lowest watt per TB	• 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	• 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 (referred to as an I/O context), it connects to a logical storage pool in the Ceph cluster. The figure below illustrates the overall Ceph architecture along with the concepts that are described in the sections that follow.



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, allowing placement on groups of servers with differing performance characteristics—allowing 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 ruleset: The CRUSH algorithm provides controlled, scalable, and de-clustered 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.

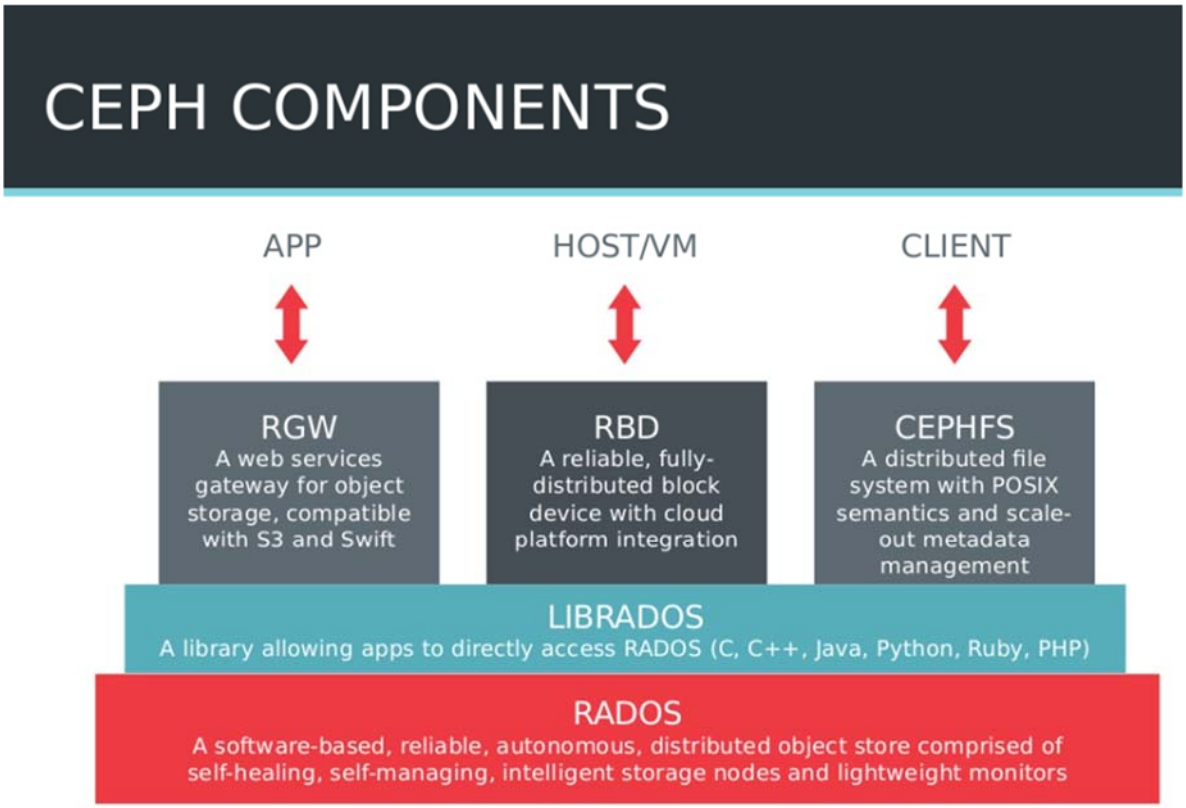
Ceph monitors (MONs): Before Ceph clients can read or write data, they must contact a Ceph MON to obtain the current cluster map. A Ceph storage cluster can operate with a single monitor, but this introduces a single point of failure. For added reliability and fault tolerance, Ceph supports an odd number of monitors in a quorum (typically three or five for small to mid-sized 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. Ceph OSD daemons roughly correspond to a file system on a hard disk drive.

SELECTING A STORAGE ACCESS METHOD

Choosing a storage access method is an important design consideration. As discussed previously, all data in Ceph is stored in pools—regardless of data type. The data itself is stored in the form of objects by 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



Writing and reading data in a Ceph storage cluster is accomplished by 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: Bucket-based 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).

Storage access method and data protection method (discussed later in this technical white paper) are interrelated. For example, Ceph block storage is currently only supported on replicated pools, while Ceph object storage is supported on either erasure-coded or replicated pools. The cost of replicated architectures is categorically more expensive than that of erasure-coded architectures because of the significant difference in media costs.

SELECTING A 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.

Ceph object storage is supported on either replicated or erasure-coded pools. Ceph block storage is typically configured with 3x replicated pools and is currently not supported directly on 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 meeting performance requirements.

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 by 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.

For more information on Ceph architecture, see the Ceph documentation at docs.ceph.com/docs/master/architecture/.

TEST SETUP & METHODOLOGY

This section describes testing of Red Hat Ceph Storage on a Dell EMC DSS 7000 test bed, including all relevant testing steps and procedures. The following subsections cover:

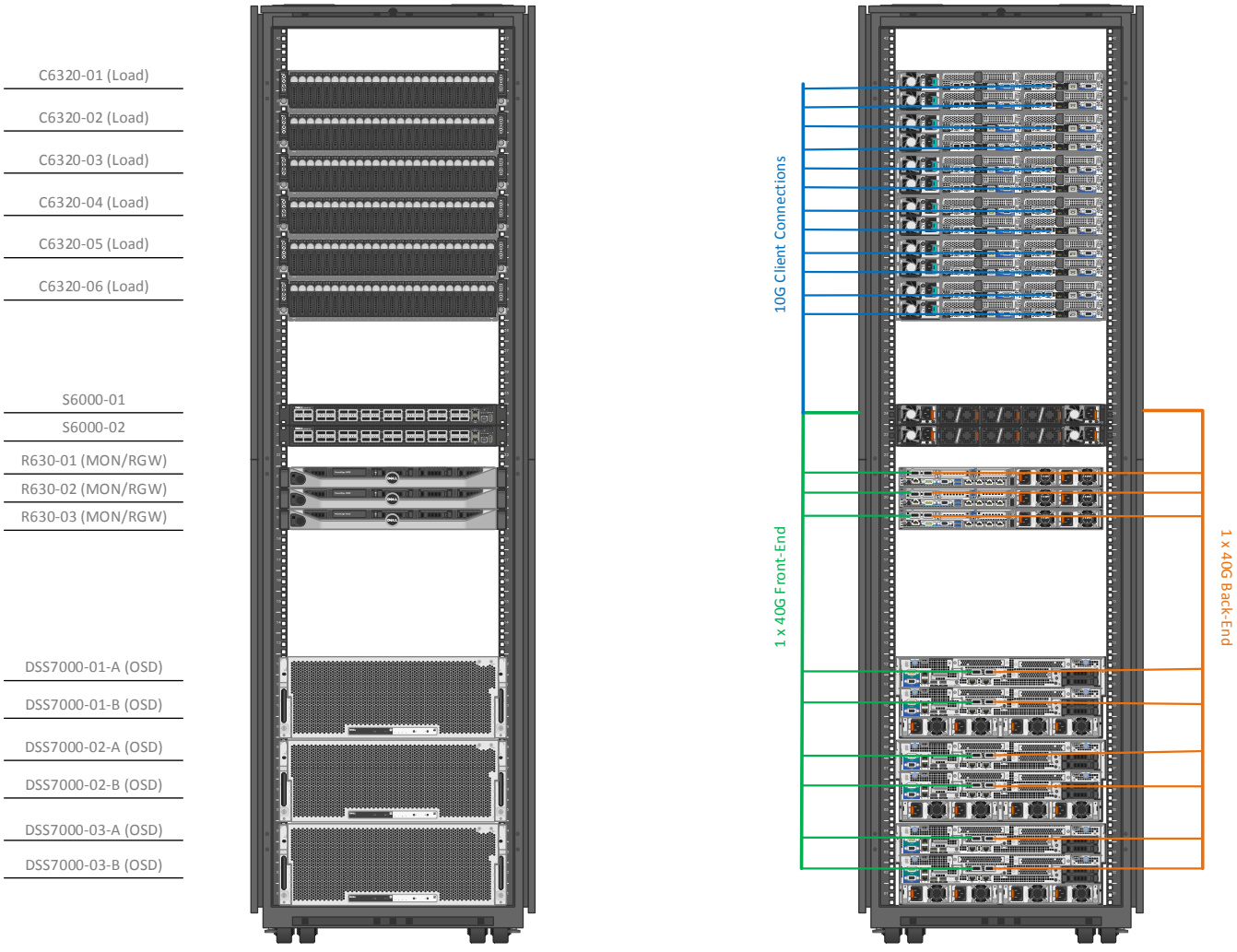
- Test bed hardware configuration
- Installation of Red Hat Ceph Storage software
- Benchmarking procedure

PHYSICAL SETUP

The figure below illustrates the layout of Red Hat Ceph Storage on Dell EMC DSS 7000. The benchmarking test bed consists of six Ceph storage nodes based on the DSS 7000 servers with forty-five 3.5" drives each. These serve as the OSD tier. The MON servers are based on three Dell EMC PowerEdge R630 servers. The load generators are based on Dell EMC C6320 servers, for a total of twenty-four clients that execute various load patterns.

Each Ceph Storage node and MON server has two 40GbE links. One link is connected to the front-end network shown below. The other link is connected to the back-end network. The load generation servers have a single 10GbE link connected to the front-end network. The ToR switching layer is provided by Dell EMC S6000 Ethernet switches.

The subnet configuration covers two separate IP subnets, one for front-end Ceph client traffic (in orange) and a separate subnet for the back-end Ceph cluster traffic (in blue). A separate 1 GbE management network is used for administrative access to all nodes through SSH, that is not shown in the Figure 5.



HARDWARE AND SOFTWARE COMPONENTS

The tables below provide details on the test bed hardware and software configurations.

Test Bed Details			
Ceph tier	OSD	MON/RGW	CLIENT
Platform	Dell EMC DSS 7000	Dell EMC PowerEdge R630	Dell EMC PowerEdge C6320
CPU	2x Intel Xeon E5-2680 v4 2.4GHz	2x Intel Xeon E5-2650 v4 2.2 GHz	2x Intel Xeon E5-2680 v3 2.5GHz
Memory	8x 32 GB 2400 MHz DDR4	8x 16 GB 2133MHz DDR4	4x 32 GB 2133MHz DDR4
Network	1x Mellanox CX3 Dual Port 40GB	1x Mellanox CX3 Dual Port 40GB	1x Intel X520 Dual Port
Storage	Broadcom MegaRAID SAS 9361-8i / 1 GB Cache 45x: SEAGATE 6 TB SATA 2x: Intel DC S3510 SSD 120 GB SATA (Boot) 2x Intel DC P3700 SSD 2TB NVMe (Journal)	PERC H730 Mini / 1 GB Cache 6x SEAGATE 500 GB SAS (ST9500620SS)	1x: Intel DC S3510 SSD 200 GB SATA (Boot)

Dell EMC network switch configuration	
Layer	Access switch
Platform	Dell EMC S6000
Ports	32x 40 GbE QSFP+

OSD to Journal Ratio [drives]	**45+2**
OSD node configuration	45+2
HDDs configured as OSDs	45
HDD RAID mode	Single-disk RAID0
NVMe SSDs configured as Ceph journals	2
Network	1x 40 GbE Front-End 1x 40 GbE Back-End

Ceph version	Red Hat Ceph Storage 2
Operating System	Red Hat Enterprise Linux 7.2
Tools	Ceph Benchmarking Tool (CBT)
Driver Versions	Mellanox: 3.3-1.0.0.0 MegaRAID SAS 9361-8i: 06.811.02.00

Server configuration	DSS7000 **45+2, 3xRep**	DSS 7000 **45+2, EC3+2**
OS disk	2x 120 GB 2.5" SSD	2x 120 GB 2.5" SSD
Data disk type	HDD 7.2K SATA 6Gbps, 6TB	HDD 7.2K SATA 6Gbps, 6TB
HDD quantity	45	45
Number of Ceph write journal devices	2	2
Ceph write journal device type	Intel P3700 NVMe AIC	Intel P3700 NVMe AIC
Ceph write journal device size (MB)	8192	8192
Controller model	MegaRAID SAS 9361-8i, 1 GB Cache	MegaRAID SAS 9361-8i, 1 GB Cache
PERC Controller configuration for HDDs	Single Drive RAID 0	Single Drive RAID 0
Raw capacity for Ceph OSDs (TB)	1440	1440

While the DSS 7000 provides a great deal of flexibility in the layout and configuration of IO subsystems, the choice was limited to the configurations listed above based on performance data of different configuration variations tested during the baselining process.

SYSTEM PERFORMANCE TUNING

Performance Tuning with sysctl.conf

For Mellanox 40G CX3 adapters optimal values for sysctl.conf were determined by following
http://www.mellanox.com/related-docs/prod_software/mlnx_en_reverse_settings.conf

The values implemented are

- `net.ipv4.tcp_timestamps=1`
- `net.ipv4.tcp_sack=1`
- `net.core.netdev_max_backlog=1000`
- `net.core.rmem_max =131071`
- `net.core.wmem_max =131071`
- `net.core.rmem_default=126976`
- `net.core.wmem_default=126976`
- `net.core.optmem_max=20480`
- `net.ipv4.tcp_rmem=4096 87380 174760`
- `net.ipv4.tcp_wmem=4096 16384 131072`
- `net.ipv4.tcp_mem=196608 262144 393216`

Disk Drive Performance tuning

The read ahead setting for all rotating drives were modified to using a udev script

/etc/udev/rules.d/99-hdd.rules

```
# Setting specific kernel parameters for a subset of block devices
        SUBSYSTEM=="block", ATTRS{vendor}=="AVAGO*", ACTION=="add|change", KERNEL=="sd[a-z]",
        ATTR{bdi/read_ahead_kb}="4096"

        SUBSYSTEM=="block", ATTRS{vendor}=="AVAGO*", ACTION=="add|change", KERNEL=="sda[a-z]",
        ATTR{bdi/read_ahead_kb}="4096"
```

Disk Drive Performance tuning

MTU was set to 9000 for all network interfaces on OSD, MON and client nodes.

DEPLOYING RED HAT ENTERPRISE LINUX (RHEL)

Red Hat Ceph Storage is a software-defined object storage technology which runs on RHEL. Thus, any system that can run RHEL and offer block storage devices is able to run Red Hat Ceph Storage. RHEL was installed by using the standard installation process as recommended by Red Hat. For additional information, please see https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Installation_Guide/.

CONFIGURING THE DELL EMC SERVERS

The Dell EMC DSS 7000 and Dell EMC PowerEdge R630 servers are configured with BIOS settings set to default with the iDRAC set for shared mode with Gb LOM 1.

The RAID controllers are configured using the standard Broadcom MegaRAID StorCLI

The HDD are configured as a single drive RAID 0 using the WB and NORA options.

DEPLOYING RED HAT CEPH STORAGE 2

In production environments, Red Hat Ceph Storage can be deployed with a single easy-to-use Ansible-based installer. For the purposes of this benchmarking, Red Hat Ceph Storage was deployed using the Red Hat Ansible playbook.

Ceph-ansible is an easy to use, end-to-end automated installation routine for Ceph clusters based on the Ansible automation framework. Relevant for this benchmarking process are mainly two configuration files in the form of Ansible variable declarations for host groups. Predefined Ansible host groups exist to denote certain servers according to their function in the Ceph cluster, namely OSD nodes, Monitor nodes, RADOS Gateway nodes, and CephFS Metadata server 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.

In this benchmark, Ceph MON and RGW roles are hosted side-by-side on the PowerEdge R630 servers, although no RGW tests were performed for this paper.

PERFORMANCE BASELINING

Before attempting benchmark scenarios that utilize higher-layer Ceph protocols, it is recommended to establish a known performance baseline of all relevant subsystems, which are:

- HDDs and SSDs (SATA + NVMe)
- Network (40 GbE Front-End and Back-End)
- Network (10 GbE Client)

The IO-related benchmarks on storage and network subsystems will be tied into direct reference of vendor specifications. As such, the following baseline benchmarks have been conducted:

Subsystem	Benchmark Tool	Benchmark Methodology
Network	iperf-2.0.8	Single TCP-Stream Benchmark All-to-All
SATA HDD	fio-2.14-3-gd2204	4K random read/write and 4M sequential read/write
NVMe SSD	fio-2.14-3-gd2204	4K random read/write and 4M sequential read/write

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.

Server Type	**DSS 7000** **Mellanox CX3 40G**	**PowerEdge R630** **Mellanox CX3 40G**	**C6320** **Intel X520 LOM**
DSS 7000 Mellanox CX3 40G	39.6 Gb/s	39.5 Gb/s	9.9 Gb/s
PowerEdge R630 Mellanox CX3 40G	39.5 Gb/s	39.6 Gb/s	9.91 Gb/s
C6320 **Intel X520 LOM**	9.9 Gb/s	9.91 Gb/s	9.9 Gb/s

Storage performance has been measured thoroughly in order to determine the maximum performance of each individual component.

In order to benchmark the raw performance of the various storage subsystems the test were run directly against the underlying block devices. The data is reported on a per-device average.

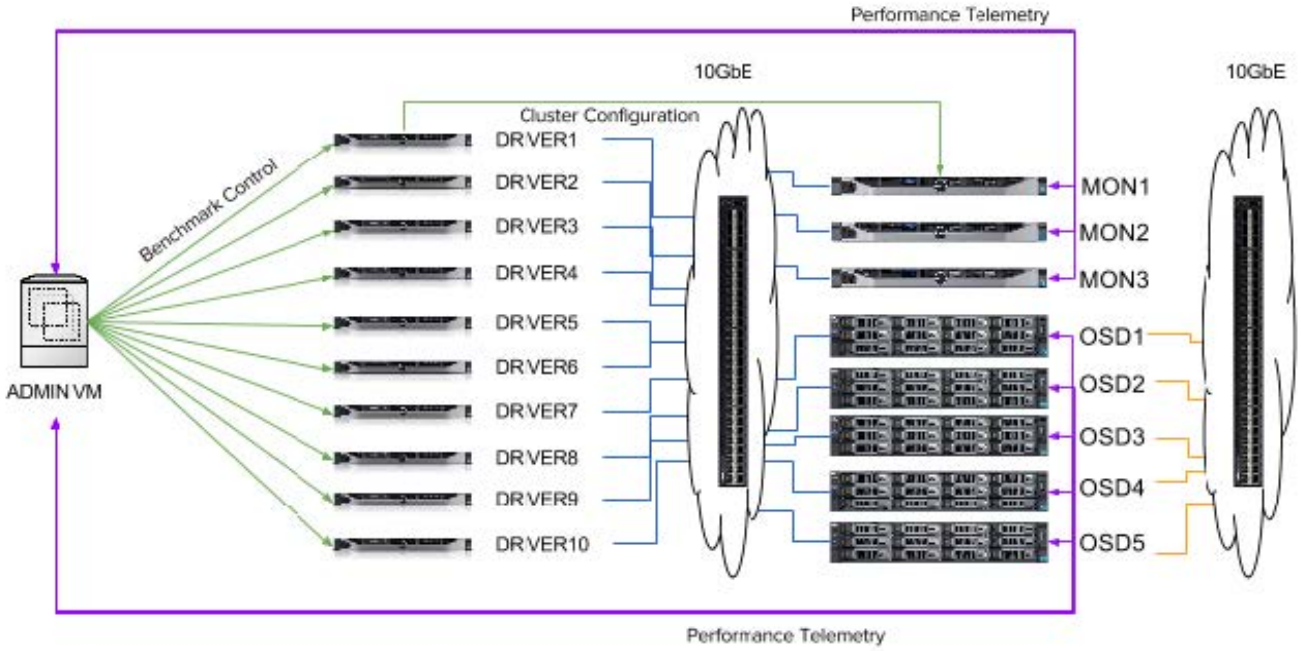
Disk Type	**OSD** **Seagate 6TB SATA**	**Journal** **Intel DC P3700 2TB**
Random Read	85 IOPS (4K blocks)	68783 IOPS (4K blocks)
Random Write	361 IOPS (4K blocks)	70889 IOPS (4K blocks)
Sequential Read	112.8 MB/s (4M blocks)	2466 MB/s (4M blocks)
Sequential Write	122.9 MB/s (4M blocks)	1908 MB/s (4M blocks)

As previously stated, this data was obtained to get a performance baseline of the systems in their current setup; not to get individual peak performance for every device tested out of context. With that said, individual components may vary in performance in other systems or when tested in isolation.

BENCHMARKING WITH CBT

For automation of the actual Ceph benchmarks, an open-source utility called the Ceph Benchmarking Tool (CBT) was used. It is available at https://github.com/ceph/cbt.

CBT is written in Python and takes a modular approach to Ceph benchmarking. The utility is able to use different benchmark drivers for examining various layers of the Ceph Storage stack, including RADOS, RADOS Block Device (RBD), RADOS Gateway (RGW) and KVM. In this paper, storage performance on the core layer RADOS is examined for which the driver in CBT uses the 'rados bench' benchmark which ships with Ceph. CBT's architecture is depicted below.



The utility is installed on Monitor Node 3. From there, it communicates with various servers in different capacities via *pdsh* as follows:

- **Head Node**: a system that has administrative access to the Ceph cluster for the purpose of creating pools, rbds, change configuration or even re-deploy the entire cluster as part of a benchmark run
- **Clients**: these are the systems which have access to the Ceph cluster and from which CBT will generate load on the cluster using locally installed tools such as fio, rados or cosbench or run VMs to access the cluster
- **OSDs/MONs**: CBT triggers performance collection with various tools such as valgrind, perf, collect or blktrace on these nodes during the benchmark run and transfers their telemetry back to the head node after each execution

The CBT configuration file syntax was used to orchestrate most of the benchmarks. CBT provides flexibility to run benchmarks over multiple cluster configurations by specifying customer *ceph.conf* files. CBT also allows the user to re-deploy the cluster between benchmark runs completely.

In this benchmark, CBT was mainly used to execute the benchmarks. The cluster deployment and configuration was provided by ansible-ceph. The setup of CBT, including necessary pre-requisites and dependencies is described on the project homepage.

The CBT job files are specified in YAML syntax, for example:

```
cluster:
  user: "root"
  head: "mon3"
  clients: [ "client1", "client2", "client3", "client4", "client5", "client6", "client7",
"client8", "client9", "client10", "client11", "client12", "client13", "client14", "client15",
"client16", "client17", "client18", "client19", "client20", "client21", "client22",
"client23", "client24", ]
  osds: ["osd1", "osd2", "osd3", "osd4", "osd5", "osd6",]
  mons:
    mon1:
      a: "192.168.1.7:6789"
    mon2:
      a: "192.168.1.8:6789"
    mon3:
      a: "192.168.1.9:6789"
  iterations: 1
  rebuild_every_test: False
  use_existing: True
  clusterid: "ceph"
  tmp_dir: "/tmp/cbt"
  pool_profiles:
    replicated:
      pg_size: 16384
      pgp_size: 16384
      replication: 3
benchmarks:
  radosbench:
    op_size: [ 4194304 ]
    write_only: False
    time: 300
    concurrent_ops: [ 128 ]
    concurrent_procs: 1
    use_existing: True
    pool_profile: "replicated"
    pool_per_proc: False
```

The file is divided in two sections: *cluster* and *benchmarks*. The first describes the cluster with the most essential data. The *user* specified here is a system user which needs to be present on all nodes and needs passwordless sudo access without the requirement for an interactive terminal. The *head* nodes, *clients* and *osds* are listed by their domain name of IP address. The MONs are specified in a syntax that distinguishes between a front-end and back-end network for Ceph. This is not used further here as the cluster setup is not done via CBT. This is expressed with the *use_existing* parameter set to *true*. The clusterid is provided based on what is described in the *ceph.conf* file. The *tmp_dir v*ariable specifies a directory on all the nodes that CBT access under *user* in which intermediate data is stored, mostly consisting of benchmark telemetry. The *pool_profiles* is a YAML list item which allows the user to employ different RADOS pool configurations referred to by name in the benchmark descriptions.

benchmarks enlists various benchmark runs (the amount of repeated execution is specified in *iterations* in the *clusters* section) that are processed in a sequential order. The name refers to a benchmark driver that ships with CBT. In this example, *radosbench* is the driver that executes low-level tests on the core librados layer of Ceph by using the *rados* bench utility. The parameters specified below are directly handed over to the rados bench call executed on the client systems, whereas list items such as *op_sizes*, *concurrent_ops* or *osd_ra* each trigger individual runs with one of their entries as the respective parameters. The description of these parameters can be found in the help output of the *rados* binary.

The same set of tunables were applied throughout the benchmarks. Ansible-ceph ships with the most recommended tunings out of the box.

A functionality that CBT currently does not provide is to sequence the repeated benchmark execution with an increasing amount of parallelism in each run. The goal of this benchmark is also to find the high water mark of the cluster's aggregated throughput, which is the point beyond which the performance increase is becoming zero or negative. Each benchmark scenario is run in 10 iterations, with the first executing the benchmark only from a single client, the second iteration with two clients, the third with three clients in parallel and so on. To achieve this, multiple instances of a benchmark job file were created; each with an increasing amount of clients. The benchmarks were then started with CBT individually by looping over a continuous set of job files.

- Common configuration
  - Write benchmarks, followed by sequential read benchmarks
  - Tested block sizes: 4M
  - Execution of each benchmark run: 5 minutes
  - 128 concurrent threads per client
  - 1 rados bench instance per client
  - Pool name: cbt-librados
- <1..24>_hosts_ec.yml
  - librados-level benchmark on an erasure-coded pool
  - erasure-coding scheme used was 4:2
- <1..24>_hosts.yml
  - librados-level benchmark on a replicated pool, replication factor 3

In each benchmark, these files are called with CBT in a loop.

CAUTION

When executing multiple instances of CBT subsequently in a loop, as in this benchmark, it is important to note that CBT will delete any existing pool with the same name. This is an asynchronous process that triggers purging object structures on the backend file store. While the command 'ceph osd pool delete' returns, instantly a potential long-running IO-intensive process on the backend is started which may collide with IO issued by a subsequent benchmark run.

It is crucial to either wait for this process to finish before starting a new benchmark, or omit the pool deletion to avoid skewed benchmark results.

In this benchmark, CBT has been modified to implement a new parameter that will cause CBT to skip deletion of existing pools at the beginning of a new benchmark.

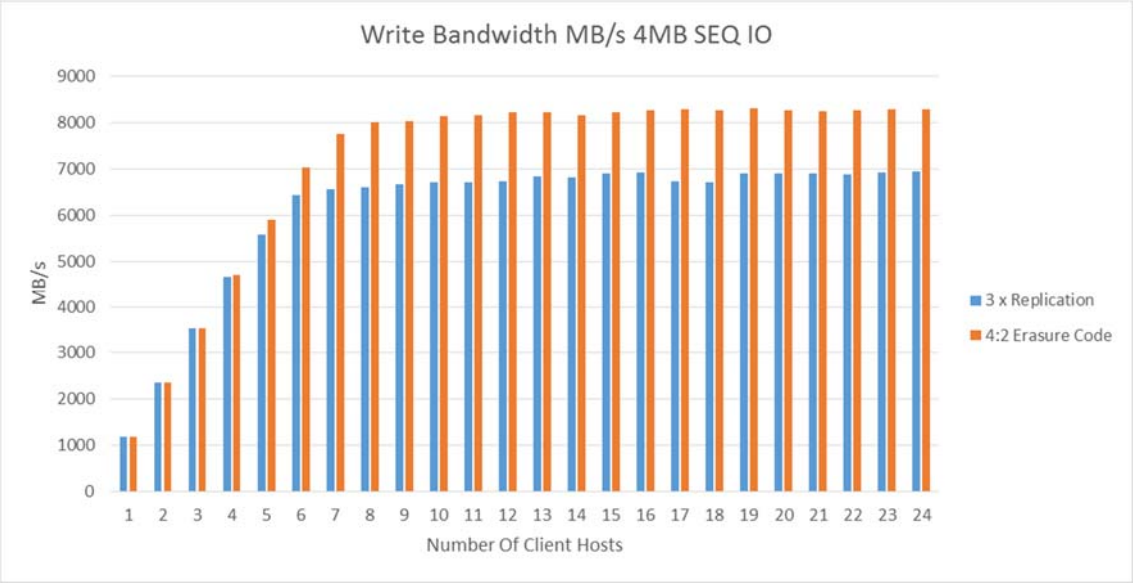
BENCHMARK TEST RESULTS

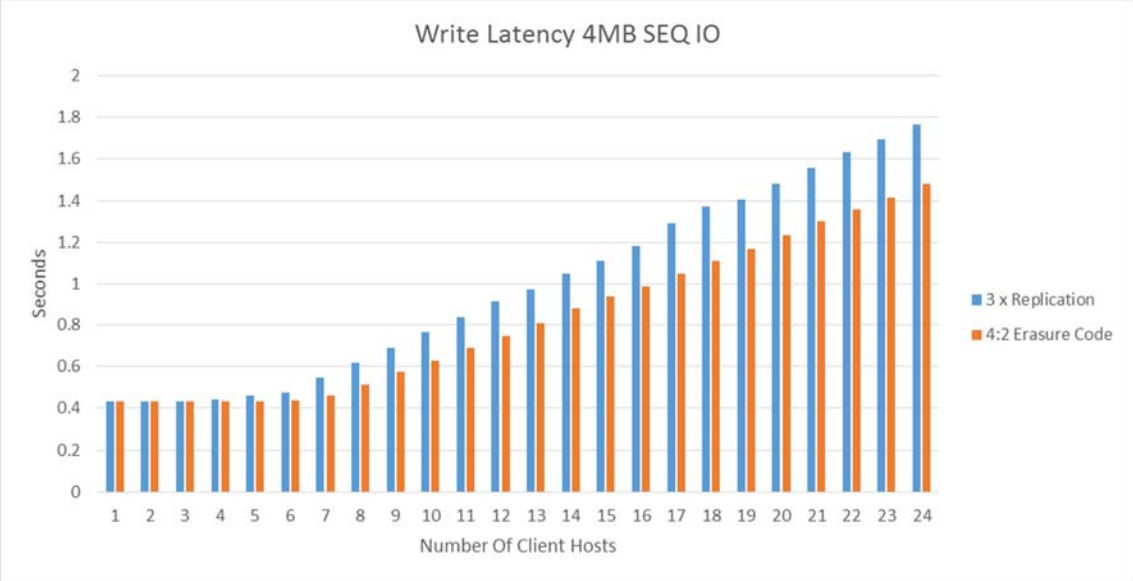
Many organizations are trying to understand how to configure hardware for optimized Ceph clusters that meet their unique needs. Red Hat Ceph Storage is able to run on a myriad of diverse industry-standard hardware configurations but designing a successful Ceph cluster requires careful analysis of issues related to application, capacity, and workload. The DSS 7000 configuration presented here is primarily aimed at high-capacity object storage since nodes with large drive counts present significant failure domains.

SYSTEM WRITE PERFORMANCE

This test compares the write throughput & write latency of the system when configured for 3x Replication and 4:2 Erasure Coding. In all graphs, MB/s reported is aggregate throughput, as measured from the cumulative client workload perspective.

- Writes: The Ceph erasure-coded configurations generally yield higher write-throughput compared to replicated configurations, because there is lower write amplification with erasure-coded writes (less MB written).

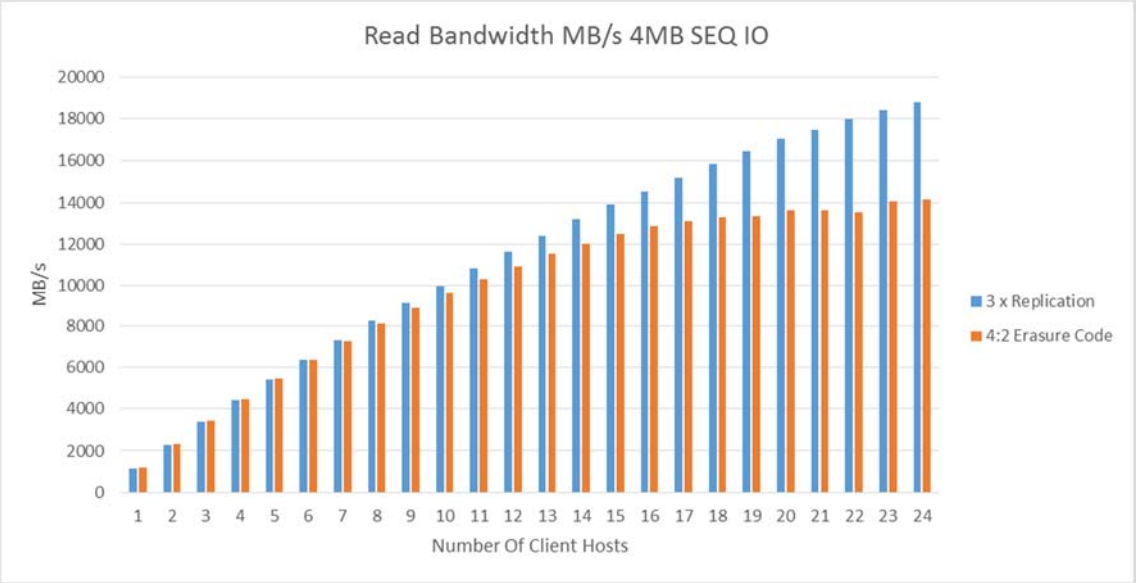


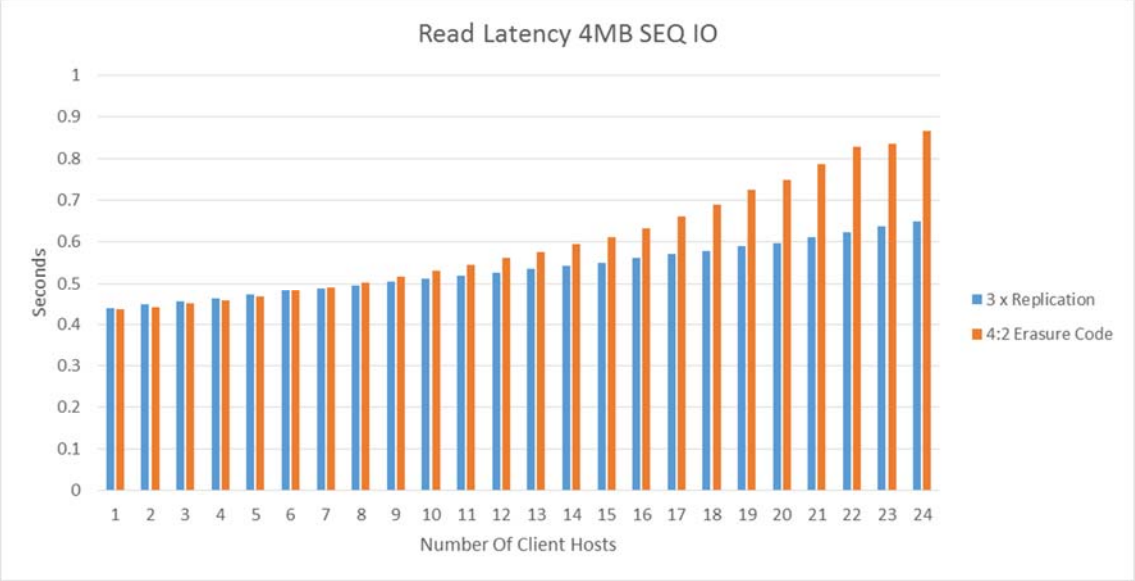


SYSTEM READ PERFORMANCE

This test compares the read throughput and read latency of the system when configured for 3x Replication and 4:2 Erasure Coding.

- Reads: The Ceph replicated configurations generally yield higher read-throughput and lower read latency compared to erasure coded configurations. The Read Bandwidth graph illustrates that, at ~19,000 MB/s, 24 client hosts are unable to saturate this 3x replicated pool cluster. With an erasure-coded pool cluster, ~17 client hosts are able to saturate this cluster.





DELL EMC SERVER RECOMMENDATIONS FOR CEPH

Ceph operators frequently request simple, optimized cluster configurations for different workload types. Common requests are for throughput-optimized and capacity-optimized workloads. Based on extensive testing by Red Hat and Dell EMC on various server configurations, this matrix provides general guidance on sizing Ceph clusters built on DSS 7000 servers.

Storage Capacity	**Large**
Cluster Capacity	1.5 PB+
Throughput-Optimized	NA
Cost/Capacity-Optimized	> 3x DSS 7000 (12U)
	2x server/4U chassis
	45x 8 TB HDD
	2x HHHL AIC SSD
	1x 40 GbE

CONCLUSIONS

After testing different combinations of Red Hat Ceph Storage on Dell EMC DSS 7000 servers to provide a highly-scalable enterprise storage solution, the following conclusions were made:

- The 3x replication configurations provided high throughput for read operations because the erasure-coded reads have to reassemble data objects from the erasure-coded chunks. However, the erasure-coded configurations demonstrated high throughput for write operations at a lower price due to less write amplification.
- Replication mode yielded better performance for read operations while the erasure-coded mode proved better for write operations.
- When used with Ceph Storage, Dell EMC and Red Hat recommend the usage of single-drive RAID0 mode on DSS 7000.

REFERENCES

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

- Red Hat Ceph Storage Hardware Guide: https://www.redhat.com/en/resources/hardware-selection-guide-ceph

  http://ceph.com/
  http://docs.ceph.com/docs/master/architecture/