Technical white paper

Ceph on HP ProLiant SL4540 Gen8 Servers

Open source object storage for unstructured data

Table of contents

Executive summary

Traditional file and block storage architectures are being challenged by the explosive growth of data, fueled by the expansion of Big Data, unstructured data, and the pervasiveness of mobile devices. Emerging storage architectures like object storage can help businesses deal with these trends, providing cost-effective storage solutions that keep up with capacity growth while providing the service-level agreements (SLAs) to meet business and customer requirements.

Enterprise-class storage subsystems are designed to address storage requirements for business-critical transactional data latencies. However, this may not be an optimal solution for unstructured data and backup/archival storage. In these cases, enterprise-class reliability is still required, but massive scale-out capacity and lower solution investment are more important than minimal latency.

Object storage software solutions are designed to run on industry-standard server platforms, offering lower infrastructure costs and scalability beyond the capacity points of typical file server storage subsystems. Ceph running on HP ProLiant hardware is a comprehensive and cost-effective object storage solution for addressing scale-out storage needs.

HP hardware combined with Ceph on Linux delivers an open source object storage solution that:

- Has software capable of scaling from dozens of terabytes, to exabytes of data and billions of objects
- Lowers upfront solution investment and total cost of ownership (TCO) per gigabyte
- Provides enterprise-class infrastructure monitoring and management
- Does not require cluster software licensing as the cluster is scaled
- Provides a single software-defined storage (SDS) cluster with block, object, and file access
- Uses open source, minimizing concerns about vendor lock-in, and increasing flexibility of hardware and software choice
- Can integrate into an OpenStack deployment

An ideal storage solution

CTOs and solution architects typically look for storage solutions that can handle the rapid growth of unstructured data, cloud, and archival storage while controlling licensing and infrastructure costs. This paper assumes knowledge of enterprise data center administration challenges and familiarity with data center configuration and deployment best practices, primarily with regard to storage systems. It also assumes the reader appreciates the challenges and benefits open source solutions can bring, especially for early adopters of object storage.

This white paper describes testing performed by HP in March 2014.

Introduction

This white paper describes a Ceph cluster deployed on HP hardware. It details why and how to build a Ceph cluster with HP hardware to solve unstructured, cloud, and backup/archival storage problems. The key reasons the reader should care about this are:

- Object storage is a better solution for unstructured data than traditional storage alone
- The right solution needs the right platform—'white box' hardware doesn't meet enterprise needs at scale

Object storage is architected for the characteristics and use of Big Data to remove scaling limitations. As implemented by Ceph, object storage is an SDS layer that federates traditional file and block storage on industry-standard Linux servers. This provides a way to scale out massively for Big Data needs at lower costs than SAN/NAS business-critical storage targets.

HP hardware is the right platform for a large-scale object storage cluster because it provides better TCO for operating and maintaining the hardware than 'white box' servers. HP provides:

- Platform management tools that scale across data centers
- Server components and form factors that are optimized for enterprise use cases
- Hardware platforms where component parts have been qualified together
- A proven support infrastructure

Clusters built with 'white box' servers work for business at small scales, but as they grow, the complexity and cost make them less compelling than enterprise-focused hardware. With 'white box' solutions, IT has to standardize and integrate platforms and supported components themselves. Support escalation becomes more complicated. Without standardized toolsets to manage the hardware at scale, IT must chart their own way with platform management and automation. Power consumption and space inefficiencies of generic platform design also limit scale and increase cost over time.

The result is IT staff working harder and the business spending more to support the quantity and complexity of a 'white box' hardware infrastructure. The lowest upfront cost does not deliver the lowest total cost or easiest solution to maintain.

Sample reference configuration summary

This Ceph cluster is shown at a high level to give the reader context; it's based around storage on the HP ProLiant SL4540 Server, which is purpose-built for Big Data. The single rack sample cluster contains:

- Five 2 x 25 HP ProLiant SL4540 Gen8 Server chassis, with 3 TB drives and SSD Journals
- Three HP ProLiant DL360p Gen8 Server chassis
- Ubuntu 12.04.03 LTS. Ubuntu is the OS best supported by Ceph software today, and the long-term support release is most appropriate for an enterprise environment
- Ceph running the Dumpling (v0.67) release, which is the most current and stable Ceph LTS release at the time of this testing
- 10 GbE Networking running on HP 5900AF switches, carrying object data traffic
- 1 GbE Networking running on an HP 2920 switch, carrying HP Integrated Light-Out (iLO) and corporate management traffic
- Rack and power components

In this configuration the HP ProLiant SL4540 Gen8 Servers are 'object storage nodes'; these are servers where scale-out storage hard drives reside. The HP ProLiant DL360p Gen8 Servers are 'management nodes' for the cluster. The HP ProLiant DL360p Gen8 Servers provide the part of the solution that maintains cluster state and object gateways to access the cluster storage through S3/Swift REST application programming interfaces (APIs). Given a RESTful interface, traffic generators can come from all kind of clients but in this test a benchmark tool run on x86 Linux boxes provides sample workload.

Overview

Business problem

Businesses are looking for better and more cost-effective ways to manage their exploding data storage requirements.

In recent years, the amount of storage required for businesses has increased dramatically. Exploration data from oil and gas, patient medical records, user- and machine-generated content, and many other data types generate massive amounts of data per day. Simultaneously, businesses are dealing with a shift from tape- to disk-based backup. Cost-per-gigabyte and ease of retrieval are important factors for choosing a solution that can scale quickly and economically over many years of continually increasing capacities and data retention requirements.

Many organizations still need to manage much—or all—of that data in-house. Regulations and privacy considerations can make offsite storage impractical or impossible. Hosting on a public cloud may not meet cost or data control requirements in the long term; the performance and control of on-premise equipment still offers real business advantages.

Organizations that have been trying to keep up with data growth using traditional file and block storage solutions are finding that the complexity of managing and operating them has grown significantly—as have the costs of storage infrastructure.

Typical architectures vs. object storage

Storage solutions designed for traditional IT tasks are not optimal for petabyte-scale unstructured data

Typical architectures often struggle to meet business SLAs when applied to petabyte scale unstructured and archival data. In addition, with traditional storage solutions, it's possible to pay for features that aren't needed, and achieve less flexibility, scale, and reliability than required by the SLA.

Here are some ways traditional thinking falls short when architecting a solution to serve unstructured data at massive scale.

Architectural and cost mismatches

- File and block storage methods that make sense for structured data impose unnecessary overhead for unstructured data, particularly at large scale. Traditionally, businesses buy block storage optimized for classic data access cases, like database workloads and file systems. These solutions have the ability to support high IOPS and heavy, concurrent write load. However, unstructured and archival data is often written just once. Bandwidth and storage capacity are much more important for unstructured and archival data than low latency. Traditional storage means paying for drive classes and features an unstructured use case may not need.
- When trying to drive the lowest cost per GB, tape immediately comes to mind. For many Big Data use cases, worst case latency of tape-based storage falls outside of the required latency behaviors for data access. Unstructured and archival data may sit dormant for a while but need to be available quickly—with maximum latency times measured in seconds instead of minutes. Where tape latencies are acceptable, many enterprises don't want to manage tape storage for onsite data.

Gaps in reliability, manageability, and scalability

- Storage systems designed for smaller-scale, single-site deployments are often not capable of delivering the overall reliability and data durability necessary to support complex, multi-site scale-out configurations.
- Many existing storage solutions are a challenge to manage and control at massive scale. Management silos and user interface limitations make it harder to deploy new storage into business infrastructure.
- Unstructured deployments can accumulate billions of objects and petabytes of data. File system limits on count and size of files, and block storage limits on the size of presented block devices become significant connection management and deployment challenges.

Why object storage technology

Businesses need an architecture that's more scalable, and provides an easier way to manage and access data. The enterprise also still requires availability and access control, even if the performance requirements are different than those of traditional storage architecture.

Object storage is designed for the scale, characteristics, and requirements of unstructured data

By creating an interface that isn't encumbered with design restrictions of file and block, but is optimized for unstructured data, it's possible to create a cluster architecture that breaks out of typical scale storage architectural drawbacks.

Object storage architecture details

Object storage allows the storage of arbitrary-sized "objects" using a flat, wide namespace where each object can be tagged with its own metadata. This simple architecture makes it much easier for software to support massive numbers of objects across the object store. The APIs provided by the object storage gateway add an additional layer above objects—called 'containers' (Swift) and 'buckets' (S3)—to hold groupings of objects.

To access the storage, a RESTful interface is used to provide better client independence and remove state tracking load on the server. HTTP is typically used as the transport mechanism to connect applications to the data, so it's very easy to connect any device over the network to the object store.

The I/O interface is designed for static data. There are no file handles, concerns for locking, or reservations on objects. An S3 or Swift API object I/O translates to an HTTP PUT (write) for the entire object, HTTP GET (read), or HTTP DELETE. Along with the flat structure, it's much easier for the storage architecture to support client concurrency because write concurrency doesn't exist. If multiple clients attempt to write to the same object, one version will 'win'; the entire resulting object will be coherent with a given client object PUT. This may not be easy to predict, so what simplifies storage architecture could impact client software.

Object storage commonly includes multi-tenancy with access keys and ACLs for storage. With a metadata-rich focus, object storage is built around 'what' is in data rather than where it's located. That means that work to guarantee enterprise availability—sites, replica counts, etc.—stays in the cluster. The client code is focused on the data context.

At the core of the object storage concept is the way clients leverage a (relatively) flat namespace, metadata tags on objects, and the RESTful interface. Various object storage interfaces may have more or less hierarchy in the namespace, allow partial writes to existing objects (RADOS does this), or might not require client features such as access or ACLs. Because this document covers object storage access through APIs provided by the object storage gateway, HP has provided additional details specific to those interfaces.

Key solution technologies

Using industry-standard servers as cluster components gives enormous flexibility for customizing, configuring, and balancing cost for the use case (CPU per disk, storage density, network infrastructure, etc.). With massive scale, costs of the cluster building blocks add up, so choosing the right components for the task makes a difference.

It's very important for enterprise adopters to develop a roadmap for understanding and implementing a maintainable object storage solution. As an early adopter of object storage in general—and an open source solution in particular—expect to realize a cost and feature benefit for implementing object storage that can make a real difference operating at scale. But also plan for an engineering load both to support a Ceph cluster and develop code to utilize object storage.

Cluster architecture

A Ceph cluster is SDS architecture layered on top of traditional server storage. It provides a federated view of storage across multiple industry-standard servers using block storage, and traditional file systems, and does this with object storage architecture. This approach has the advantages of leveraging work and standard hardware where appropriate, while still providing the overall solution scale and performance needed. See Ceph Architecture for more details.

The core of mapping a GET/PUT or block read/write to Ceph objects from any of the access methods is Controlled Replication Under Scalable Hashing (CRUSH). It is the algorithm Ceph uses to compute object storage locations. All access methods are converted into some number of Ceph native objects on the back end.

Cluster roles

There are three primary roles in the Ceph cluster covered by this sample reference configuration.

OSD Host—The HP ProLiant SL4540 Gen8 Server has been presented as the object storage host; this is how Ceph terms the role of the server storing object data. The Ceph OSD Daemon is software which interacts with the OSD (Object Storage Disk); for production clusters there's a 1:1 mapping of OSD Daemon to logical volume. The default file system used for this sample reference configuration on an OSD is xfs, although btrfs and ext4 are also supported.

Ceph Monitor (MON)—A Ceph Monitor maintains maps of the cluster state, including the monitor map, the OSD map, the Placement Group Map, and the CRUSH map. Ceph maintains a history (called an "epoch") of each state change in the Ceph Monitors, Ceph OSD Daemons, and PGs.

Object Gateway (RGW)—An object storage interface to provide applications with a RESTful gateway to Ceph Storage Clusters. Ceph Object Storage Gateway supports two interfaces, S3 and Swift. These interfaces support a large subset of their respective APIs as implemented by Amazon and OpenStack Swift.

Value of a purpose-built enterprise hardware platform

An important part of planning Ceph cluster architecture is determining what kind of hardware it runs on. HP hardware brings value to the solution in these ways:

- Flexible compute/storage ratio—With one, two, and three compute node chassis available, you can choose the HP ProLiant SL4540 Gen8 Server model that delivers the optimal storage-to-compute ratio for your object storage access workloads.
- Converged design—The HP ProLiant SL4540 Gen8 Server delivers increased storage density at lower cost.
- Flexible I/O bay—Storage controller and network interfaces are contained within each compute node's I/O bay. 10 GbE and 1 GbE networking options ensure support for industry-standard network infrastructures.
- Power management—The SL Advanced Power Manager provides dynamic power capping and asset management features that are standard across the HP ProLiant SL line. The converged HP ProLiant SL4540 Gen8 Server chassis also yields power savings via shared cooling and power resources.
- Solution integration and data center acceptance—HP hardware described in this white paper has been qualified together. This means no work building, maintaining, and qualifying 'white box' architectures for the cluster. HP hardware can be validated with confidence.
- Enterprise support—Get dedicated solution and support resources from HP, a trusted enterprise partner. At massive scale, system failures become part of the design even with the most reliable components. Therefore, it's critical to have good support infrastructure to keep system reliability and availability at acceptable levels.
- Enterprise-class storage components—HP Smart Array controllers provide a robust storage solution within the server. HP drive carriers allow easy drive swaps and collect triage information to help the RMA process not only replace failures but avoid future problems. Issues like bad drive batches and drive firmware problems are very significant for solutions that consume and scale with large quantities of drives. HP server solutions are built to manage storage failures; HP also qualifies and supports hard drives to minimize failure impacts.
- HP Integrated Lights-Out (iLO)—HP iLO is an industry-leading embedded monitoring solution. Its agentless management, diagnostic tools, and remote support allow for entire data centers to be managed with ease.
- Enterprise-class management—HP OneView provides infrastructure management for hardware at scale. Along with all the other platform management value it brings, HP OneView links failed cluster components to the location of hardware in the infrastructure.

Open source value

Businesses that use open source value the control and cost benefit it brings
Linux and Ceph provide the kind of robust and functional open source solutions these businesses want. Tradeoffs for engineering and support vs. a normal enterprise closed-source solution make sense for them.

Control
With access to the source, a business can customize solutions as needed. They can also apply bug fixes or roll new features as needed, with the ability to see exactly what's changing. There's no concern about the provider of a solution going away and making a solution unsupportable.

As an open standard, a Ceph cluster is not tied to particular hardware. This means expansion or refresh of cluster hardware is not locked in to any vendor—choose the hardware that's the right fit for the business case and solution parameters. The HP ProLiant SL4540 Gen8 Server was designed with storage density and compute ratios that match unstructured data retention and processing needs. Therefore, it's a good match for Ceph cluster storage design.

Cost
People not familiar with open source may believe that it's free because it's freely available. There are, of course, engineering costs for configuring and maintaining open source. Additionally, there are many valuable commercial software and support solutions built on top of open source. However, closed source clusters can add significant cost per server for licensing and support, which adds up at massive scale. Because there isn't a paid license for open source software, adding nodes doesn't add upfront license costs. Also, building up expertise on the solution in-house can pay off by reducing the operating expense (OPEX) required to support each cluster node. A proper analysis of the costs and scope of supporting an open source solution can realize significant savings at scale.

Ceph value

Active community
It's important that the community supporting an open source solution and code base is active. Ceph fits that description; in 2013 alone it grew its author pool from 103 to 203 and accepted major source contributions from significant storage industry players. The community held the inaugural Ceph Developer Summit and organized Ceph Days for education and idea exchange. Inktank is the company delivering Ceph, and they have a goal to drive the widespread adoption of SDS with Ceph and help customers scale storage to the Exabyte level and beyond in a cost-effective way.

Enterprise solutions and support
While Ceph is in use for a variety of business cases, there's ongoing work to support the needs of enterprise deployments beyond just hardening work. If the business requires it, Inktank provides professional solution support for the cluster and professional services such as performance tuning to maximize use of cluster resources.

Inktank is also creating robust enterprise management software for Ceph. The graphical manager, named Calamari, accelerates and simplifies cluster management by showing the performance and state data needed to operate a Ceph cluster. Calamari already includes cluster management but will add support for analytics within 2014. Additionally, Ceph will add support for SNMP and hypervisors like Microsoft® Hyper-V and VMware to allow better integration of a Ceph cluster into the data center cloud environment.

Use storage that matches the needs of data
Ceph's cluster reliability allows utilizing non-enterprise-class drives for significant scale savings. If faster storage is needed, Ceph can be configured to restrict a pool to a more performant tier—particularly useful for RADOS Block Devices (RBD). With replication, data consistency, and the cluster reliability of a properly tuned CRUSH map, Ceph provides enterprise data availability and durability required at petabyte scales and beyond.

Flexible access methods

Ceph can provide many different methods of storage access within a single storage cluster; this whitepaper covers object gateway and block access but file and native RADOS methods are also available.

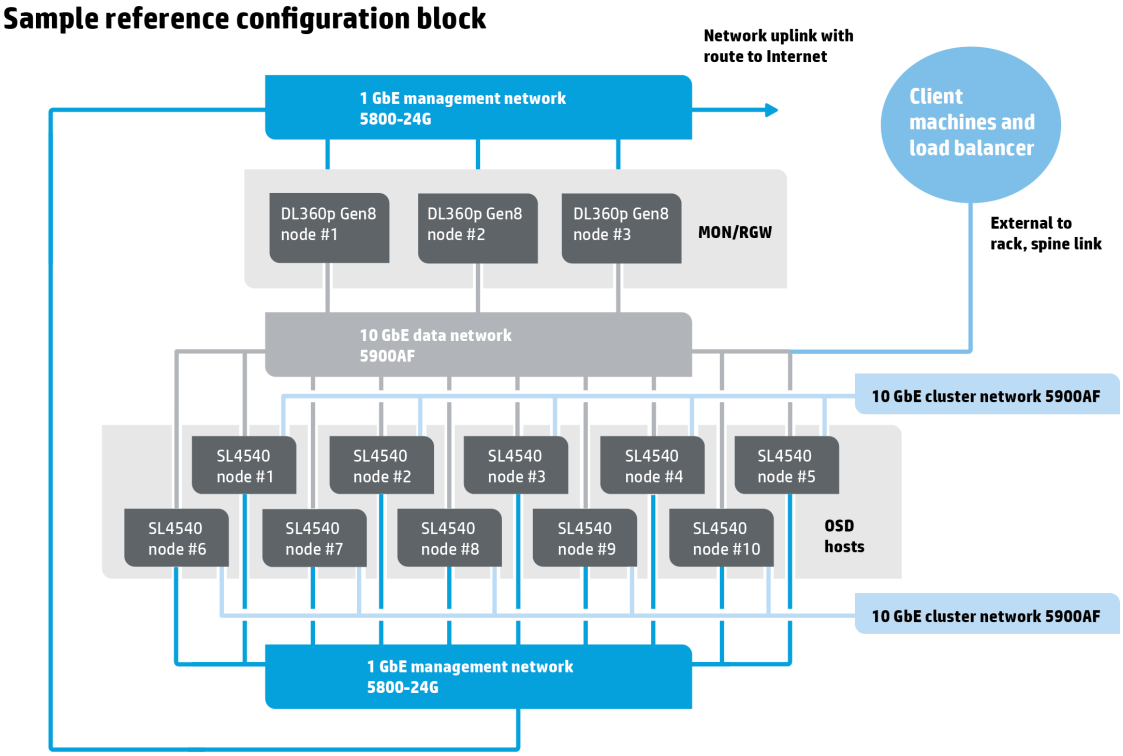
For any storage access, customers generally want methods that are supported across many storage systems. To this end, the object gateway converts S3 and Swift compatible APIs to RADOS objects. This allows existing libraries or applications that use these APIs to be leveraged rather than rewritten. So use cases like hybrid public/private cloud setups, S3 repatriation, or heterogeneous object solution environments can share and reuse more code.

Ceph also can present cluster storage with a block interface that's been supported in the Linux kernel since 2.6.35. Traditional block-focused applications and standard OS file systems can then leverage cluster storage. Within a cloud environment, RBD integrates with OpenStack Cinder/Glance and can be directly used by Linux virtual machines (VMs) themselves.

Solution diagram

This block diagram has connections to infrastructure outside of the sample reference configuration. The 1 GbE Management Network labeled in blue has an uplink to the larger management network, and the management network has a route to the Internet. Internet access allows rack management network infrastructure to reach Ubuntu and Ceph package repositories, while the uplink connects the cluster with other management servers and consoles. On the same diagram, the external link on the green labeled 10 GbE Data Network connects the cluster to client machines and load balancing.

Figure 1. Sample reference configuration block diagram



Solution components

Component choices

This section describes a more detailed reasoning behind some of the hardware and software components chosen for the sample reference configuration. Decisions made for component sizing choices in the cluster (compute, memory, storage, networking topology) are described under "Configuration guidance."

Operating system

Ubuntu is the Linux OS distribution that has been tested the most with Ceph, and Ubuntu 12.04 LTS is the most appropriate current release for enterprise use. The Long Term Support (LTS) version focuses on stability rather than including newest features. It is also the only supported cluster Ubuntu version for Ceph Calamari management software at time of writing.

HP strongly recommends having Internet available for OS install and Ceph cluster setup, especially for package management. If installing Ubuntu 12.04 completely from media, the kernel and OS packages should be updated from what initially shipped to current. It's possible to install Ceph from source code and/or set up internal package repositories.

Ceph is also supported on Red Hat and other Ubuntu distributions; reference Ceph's official documentation pages.

Switches

Top of Rack switches (HP 5900AF-48XG-4QSPF+) for Data and Replication traffic

The HP 5900AF-48XG-4QSPF+ 10 GbE high-density, ultra-low latency, TOR switch provides IRF Bonding and sFlow, which simplifies the management, monitoring, and resiliency of the network. This model has 48X 10-Gigabit/Gigabit SFP+ ports with four QSFP+ 40-Gigabit ports for ultra-high capacity connections. The high performance 10 GbE networking provides cut-through and nonblocking architecture, delivering industry-leading low latency (about one microsecond) for very demanding enterprise applications. The switch delivers a 1.28 Tbps switching capacity and 952.32 Mpps packet forwarding rate in addition to incorporating 9 MB of packet buffers.

Figure 2. HP 5900AF-48X-4QSF+ Top of Rack switch

Top of Rack switches (HP 2920 48G) for HP iLO and management

The HP 2920-48G is an ideal TOR 1 GbE switch for denser rack configurations with up to four 10 GbE uplinks, and 48 1 GbE ports. A dedicated management switch for HP iLO traffic is required for the HP ProLiant SL4540 Gen8 Server, and this also helps segment other non-cluster traffic (SSH connectivity, package updates).

Figure 3. HP 2920-48G Top of Rack switch

Both of the switches referenced are rear-facing, in that the cables for the switch are connected on the same side of the rack as the cables that are connected to the network interface cards (NICs) at the back of the HP ProLiant SL4540 Gen8 Servers.

Server selection

Within this architecture, the cluster can be scaled effectively while using the same server hardware. This section briefly talks about sample reference configuration server choices.

Management nodes

The 1U HP ProLiant DL360p Gen8 Server is a dual socket server, with a choice of Intel® Xeon® E5-2600 v2 and Intel Xeon E5-2600 processors, up to 768 GB of memory, and two expansion slots. Network connectivity can be provided through FlexibleLOM in a 4 x 1 GbE NIC configuration or a 2 x 10 GbE configuration. For storage, various configurations are available with LFF or SFF drives with an HP Smart Array P420i controller.

The HP ProLiant DL360p Gen8 Server was chosen to keep rack space requirements minimal for nodes where storage density was not the issue, but still provide good network bandwidth and compute power. An 8 SFF drive configuration is used in the sample reference configuration, but the storage on the HP ProLiant DL360p Gen8 Server is not particularly important to Ceph functionality outside of providing a reliable mirrored OS boot drive.

Figure 4. HP ProLiant DL360p Gen8 Server

Object storage nodes

The two-node configuration of the HP ProLiant SL4540 Gen8 Server consists of up to two compute nodes and a total of 50 large form factor (LFF) 3.5" hard disk drives (HDD) in the chassis. The HP ProLiant SL4540 Gen8 Server is a dual socket server, with a choice of five different Intel Xeon processors, up to 288 GB of memory, and one PCIe slot for expansion per node. Every compute node also has its own dedicated networking ports with 1 GbE and 10 GbE choices available.

The HP ProLiant SL4540 Gen8 Server was chosen as a chassis due to its focus on rack storage density, matching unstructured storage requirements. The 2 x 25 configuration specifically gives a good balance point between maximum CPU per node and colder use case storage density for tests that create a performance baseline.

Figure 5. HP ProLiant SL4540 (2 x 25) Gen8 Server

Sample reference configuration design

The sample reference configuration could have represented anything from a minimal test configuration to multiple performance optimized data centers. A bill of materials (BOM) of five HP ProLiant Gen8 SL4540 Servers and three HP ProLiant DL360p Gen8 Servers was chosen because its size is representative of enterprise data needs without being too large to be a reasonable initial deployment use case for many customers.

For raw capacity, this configuration could reach 1 PB (50 4 TB drives x 5), which is both a good conceptual scale number and a point where enterprise platform architectures make TCO sense vs. smaller 'white box' configurations. 1 PB raw of 4 TB even in 12 drive 2u boxes would still be an entire 42u rack worth (with no space for TOR switching etc.), and more standard scale-out platforms are even less rack space/port efficient.

HP's drive choice was lower cost/density but still performant midline 3 TB device. Another important storage design choice in this reference configuration is using SSD journals. Due to the architecture of Ceph's object commits significant 'PUT' performance will be gained by committing the journal and data parts of the object I/O on different devices.

The sample reference configuration fits in a single rack, but is scalable in some important ways. The rack reserves space to configure for further HP ProLiant SL4540 Gen8 Server scaling or other data center equipment. It's relatively simple to source this configuration to multiple racks by replicating elements of the BOM and distributing monitors/object gateways across the racks.

Licensing and support

The BOM lists service, support, and licenses for iLO. These are important for a scale-out solution with industry-standard servers as they provide reliability and management required to operate petabyte scale clusters and beyond. HP iLO provides the foundation for linking the hardware platform to cluster performance, along with remote hardware management. HP service and support provide expertise through setup, operation, and escalation for issues with HP provided hardware.

Inktank support on the software solution side is also recommended to protect your cluster investment. Inktank Ceph Enterprise is a subscription combining the most stable version of Ceph for object and block storage, with the Calamari graphical manager, enhanced integration tools, and support services. Inktank also provides expertise and professional services for your Ceph cluster.

Workload testing

Object and block data are the chosen focus cases for unstructured data use on Ceph. Traffic to the Ceph object gateway is using the Swift API, which has the advantage of a similar API and traffic generation test tool for both OpenStack Swift and Ceph testing. These results help set expectations for performance at a level of load and type of access, which helps scale planning and matching cluster capabilities to use case.

Workload description

Without a recognized standard for object storage benchmarking, baseline data comes from canned I/O testing. The test takes objects of varying sizes and is run to achieve a reliable average performance sample.

Test matrix configuration

The cluster is pre-seeded with 1,000 accounts: 100 containers per account and 100 objects per container. This gives a representation of I/O running on a system with used capacity. Runs operate at a fixed number of processes/threads (30) using three traffic generators for all tests.

Test matrix terminology

- A 'Suite' is all tests run for a given access method.
- A 'Pass' is all types of test at a given object/block size.
- A 'Phase' is a single type of test at an object size.
- A 'Step' is any subdivision of a Phase.

Object testing

- Test passes are done at 1 KB, 16 KB, 64 KB, 128 KB, 512 KB, 1 MB, 4 MB, 16 MB, and 128 MB object sizes.
- The account, containers, and objects being accessed by the test are not pre-seeded.
- Each tested phase has steps at 100 percent PUTs, 100 percent GETs, and then 90 percent/10 percent GETs to PUTs.
- Object count for a pass is chosen so the 100 percent PUT phase lasts at least 30 minutes.
- On the MIX step, do PUTs as step number one. Step number two is GETs to step number one objects and PUTs using new object name prefix.
- There are no object DELETEs between sizes.

This suite consists of pure write and read load tests, and then a mixed test to represent an active cluster with mostly static data being read (but some ongoing writes). As load tests are a 'warmer' use case, the MIX approximates a file hosting service load. GETs must be performed after PUTs, so file system cache impact occurs. Although object storage does not have the same SLA as traditional storage, performance benchmarking still reveals bottlenecks and resource restrictions.

Block testing

- Test phases for random I/O are 8 k read, write, and 70 percent read/30 percent write mix. Test phases for sequential I/O are 256 k read and write. All block I/O is submitted to the same 4 TB RADOS block device mapped to all three traffic generators. The sequential tests are started at offsets of 0, 1, and 2 TB on the block device.
- The RBD pool was left at default 4 M striping.
- Block I/O test passes last 30 minutes each.
- The test was set up with a level of performance that is 'reasonably' stressful to characterize the cluster, rather than for maximum performance. The iodepth was set to eight, and ioengine used was asynchronous.

These mixes are a characterization of 'real world' small block random and large block sequential loads, respectively. Tests like these are a subset of common canned test block benchmark loads and are representative of I/O on VM boot/data drive images. Unlike object testing, there isn't as much opportunity to do caching on the reads—they're done before a write phase or with random distribution across the pool. The biggest performance limiting factor here is the amount of load from the test, not an element of the cluster. The block test suite represents load on a cluster with performance headroom.

Bounding principles and choices

With a large number of variables and a lot of data to present, the test matrix was chosen as a good representation of cluster behavior under load while limiting scaling and tuning variables. This type of benchmarking won't represent production traffic, but does form a base for the reader to extrapolate from when configuring their own cluster.

Important factors to consider about the tests chosen include:

- Without a particular use case to simulate, the test standardizes on a single thread count to stress the system but not overly thrash. There's no perfect thread count across all object sizes, so the number aims for a 'good' fit.
- Traffic generators were pushed to utilize as much network bandwidth and CPU as possible. This means very few clients required to saturate resources. A production environment usually has less bandwidth per client and a higher number of clients for application load, but that's a variable HP is not currently prepared to model in a way useful to the reader.
- High bandwidth PUT tests are unlikely to be useful for colder object storage load planning.
- DELETEs are not benchmarked from a performance standpoint for a few reasons. They didn't seem as critical to system performance planning, as this class of data is purged infrequently. Under load, variance per object size was much less significant than variances for GET and PUT. They're time consuming to accurately gauge at higher object sizes since it takes so much more time writing objects than it does to acquire a significant DELETE sample.

Workload results and analysis

These cover bandwidth, IOPS, and latency data for object and block I/O tests. Object data also includes CPU usage graphs representing load on an OSD host and object gateways. I/O results are the sum of the three traffic generator client results.

General points

The analysis details will help make cluster planning decisions vs. the target workload/use case, but a few general points that can be derived from the data are:

- Reads are significantly more performant than writes at the same size.
- Writes mixed with reads have a noticeable impact on read performance.
- Object I/O maximum latency can be significant, although max latency cases are atypical.

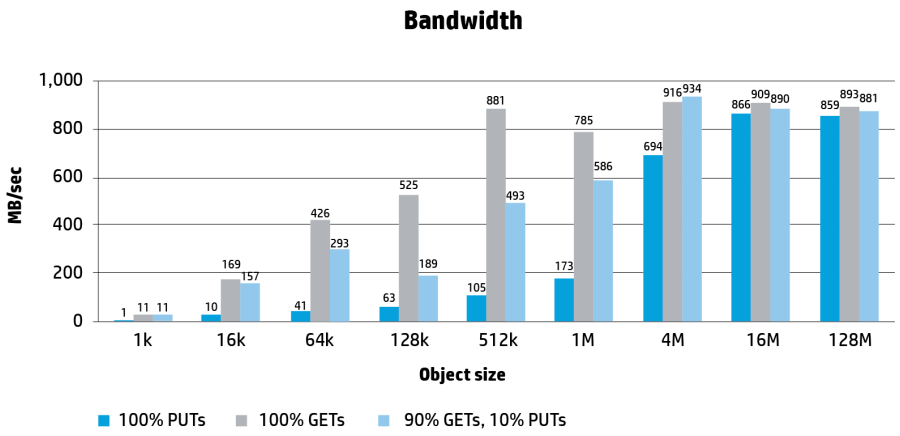
Object testing

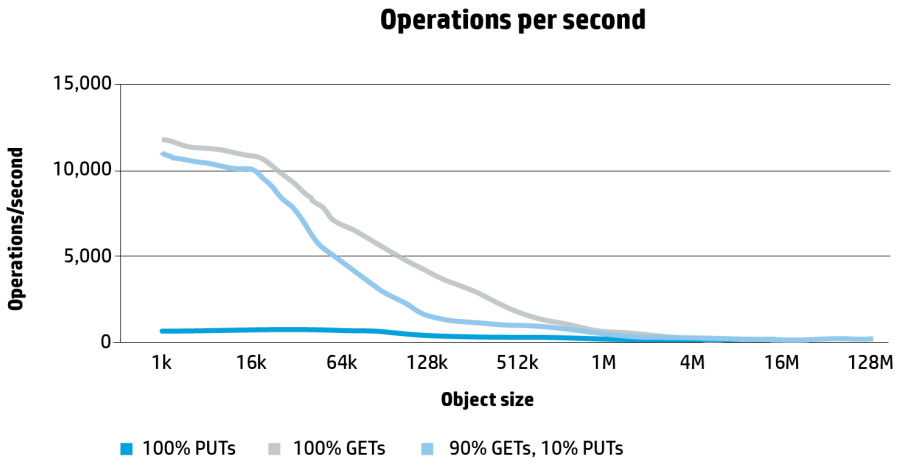
There are two I/O sizes of note in the object matrix. One is 512 k, which is typically the largest sequential I/O issued at the kernel block layer. The other is 4 M, the size of Ceph's RADOS objects in the target pools. Objects greater than 4 M submitted using the Swift API must be split into multiple RADOS objects.

While the object server listening on HTTPS is configured—and a test suite was run over SSL—the detailed results here are unencrypted traffic. Expect an additional processing load for using HTTPS at the object gateway and on the clients; the largest effect was at highest object sizes (16 M, 128 M), where average client utilization increased by a bit over 10 percent and object gateway load was up 5–8 percent on average. Peak spikes were also up significantly for HTTPS with large objects; at 128 M PUT and MIX tests rose to the low 40 percent range while GETs went from 9 percent to 17 percent peak CPU.

Object gateway test infrastructure is bottlenecking bandwidth within a single 10 GbE link; the results show about 900 MB/sec as the roll-off point for average bandwidth. Quick samples show greater peak I/O (about 1,100 MB/sec).

Bandwidth and IOPS

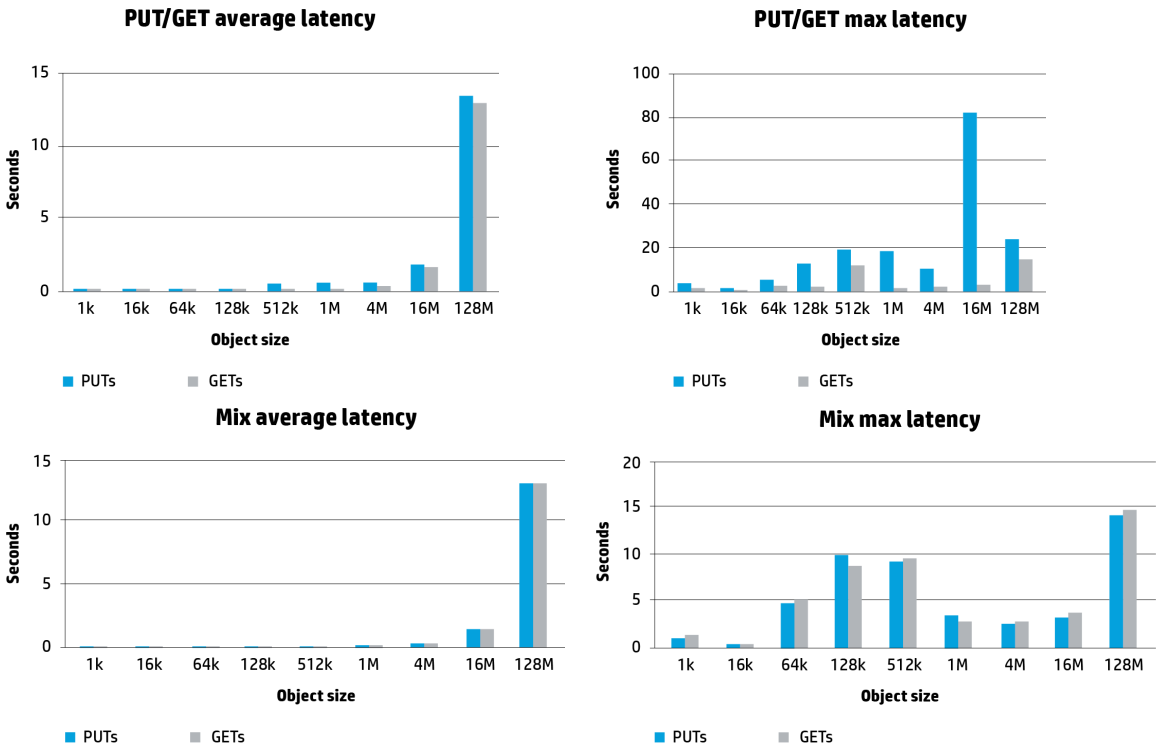




- GET ops/sec on the sample reference configuration is significantly higher than PUTs for object sizes up to the 4 M native RADOS object size. From 1 k through 512 k, the difference is 10X or more. Operation speed is particularly impacted by file system caching and lower cluster load of GETs (reads touch only one object copy, writes must commit all replicas).
- Consequently, GET bandwidth ramps a lot faster than PUT as object sizes increase in the test matrix. This means 100 percent GET bottlenecks quickly on networking in the load balancer/object gateway part of this setup. Effectively this happens around 512 k, although there's an efficiency dip at the 1 M sample.
- The MIX test has middle of the graph (64 k–1 M) results 'pulled' by interleaving PUTs significantly. None of these samples are close to a 90 percent scale factor of pure GET I/O. On the small side (1 k, 16 k), there's not much stress from GETs, and for larger objects the balancer/object gateway bottleneck prevents an accurate picture of relative GET/PUT performance.

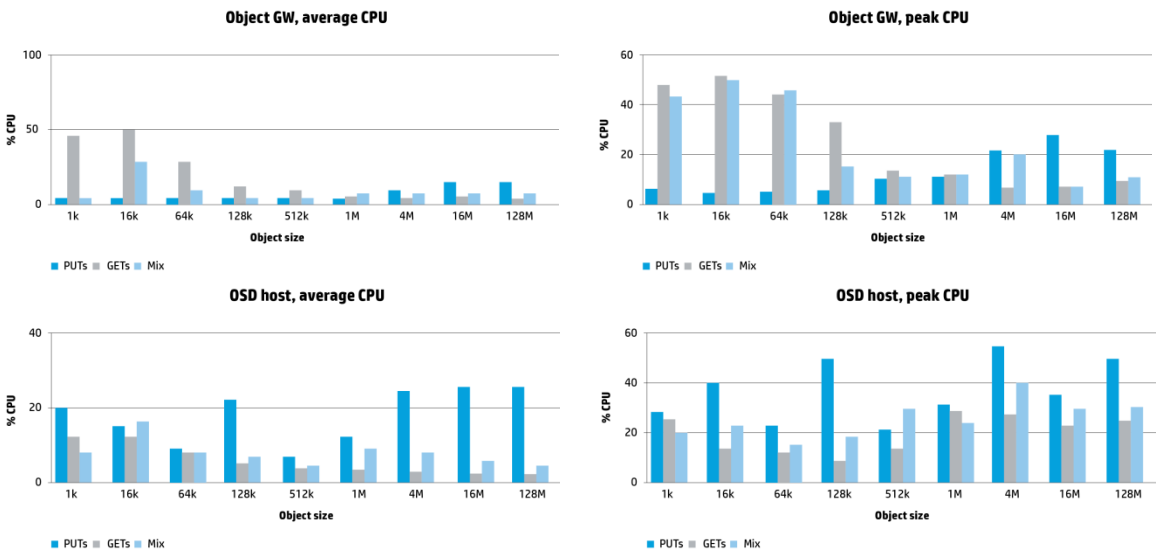
Latency

Object storage latencies are higher than typical SAN storage. Some of that is expected with the architecture (HTTP server, networking), but those factors don't cover all performance impacts. Minimum latency data for object I/O is less interesting—it's still relatively long compared to block—so those graphs are not presented.



- Average PUT latencies are in the hundreds of milliseconds, with a significant uptick around 512 k vs. smaller object sizes. One-to-many Swift I/Os that result in a number of RADOS objects being written (16 M, 128 M) are less latent than: 4 M I/O latency* (object size/4 M).
- Average GET latencies are generally faster than PUT, which agrees with the bandwidth and IOPS results above. Only the 128 M sample doesn't show much difference between PUT and GET latency.
- Maximum latency for GETs is not much worse than average latency, and is sub-20 seconds across the board. Maximum PUT latency is greater for all but the 128 M case. The very large spike at 16 M appears to be an aberration as compared with average latency, but it's important to note that max latency for object PUTs can be in the tens of seconds.
- The MIX test is a more complicated picture. The PUT and GET average latencies show almost identical tracking to each other, and are similar to the 100 percent GET/PUT test passes. Max latencies form an S curve, where the range between 64 k and 1 M spike. The 'pull' effect of PUTs show here with similar maximum latencies for GETs and PUTs, and those maximum latencies are greater than 100 percent GET but significantly less than 100 percent PUT.

CPU percentage



The results show the selected CPU doesn't go much above 50 percent even at peak, so there's plenty of CPU headroom.

GET traffic

- The object gateway shows average CPU usage highest for small objects, ramping down to fairly minimal around 1 M. Small objects are constrained by IOPS processing here. Peak CPU at the object gateway is much higher from 64 k to 512 k, settling down again in the larger object size ranges.
- On the OSD Host, the average CPU follows the same curve as the object gateway but proportionally less so since the I/Os through three object gateways are distributed across the ten nodes of the Ceph cluster. OSD Host peak CPU is interestingly different; past the 512 k mark the increased impact of I/O missing file system cache is visible.

PUT traffic

- Utilization curve on the object gateway is reversed from GETs (most CPU is at largest object sizes) and never reaches as high. Some of this is from the object gateway issuing the original I/O and waiting for the primary OSD(s) to complete the object replication. The higher bandwidth and multiple 'slices' for processing larger objects keeps the gateway busier.
- On the OSD Host, PUTs always go to disk and there's a proportionally larger amount of I/O from replication. So there's higher load, and large objects can saturate drives depending on cluster object distribution. The valley around 512 k is from maximal block I/O efficiency; 64 k is also an improved CPU efficiency point vs. very small objects.

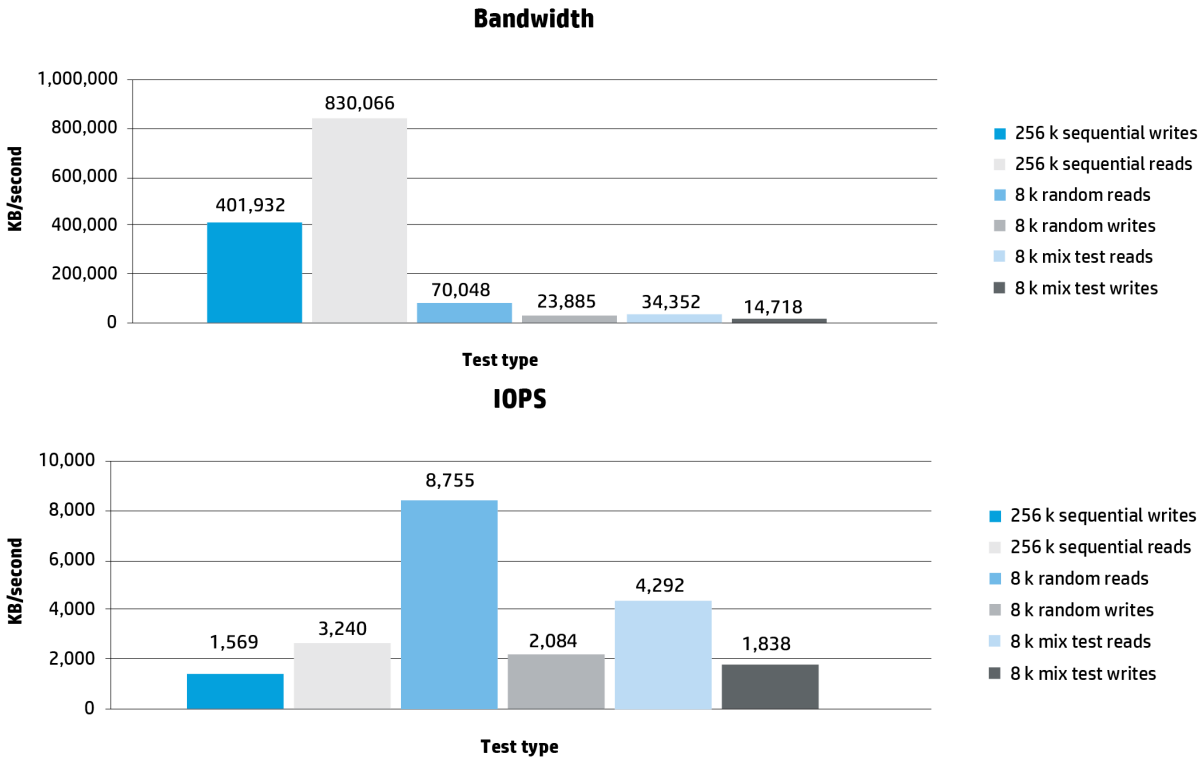
MIX traffic

- On the object gateway, average CPU roughly tracks the lowest common denominator between GETs and PUTs (although there's a noticeable spike on the 16 k object size). At peak, the load matches closer to GET—the dominating portion of the 90/10 MIX. There are two points where the MIX test peak is significantly closer to the PUT line: 128 k and 4 M.
- OSD average CPU shows a very similar tracking to GET I/O with a 16 k spike. At peak, the impact of processing PUTs keeps the MIX load around or above what's measured for 100 percent GET traffic.

Block testing

HP presents less data around RBD traffic than object I/O partly because there's more public content around tuning and performance for RBD. One reason for that is because RBD testing is easier to set up. No object gateway or object storage access code is required, and block storage benchmarking tools are easy to get and well understood. It's recommended to search for some of this other content for more detail; Mark Nelson's performance blog posts at Inktank are good places to start as are Ceph community comments around RBD performance.

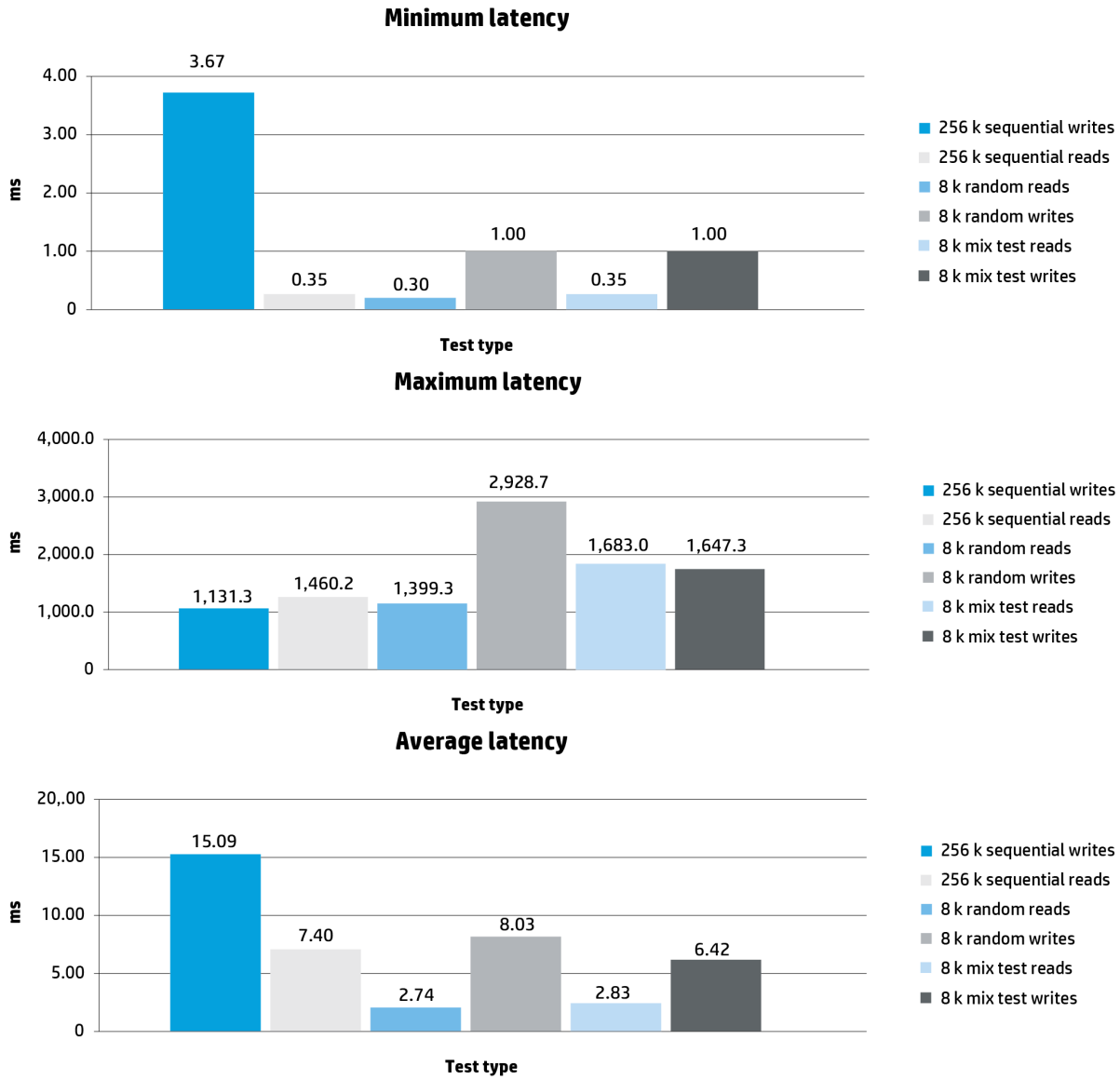
Bandwidth and IOPS



The results show a fair amount of bandwidth on sequential I/O tests; the sequential read test is getting more than 800 MB/sec with a queue depth of eight. Sequential writes perform at a bit less than half of that; this matches with expectations from the way replica traffic behaves and the lighter load in an optimal case (4 M object I/O).

Random I/O pure read peaks at 8,755 IOPS, with pure write significantly less at about 24 percent of that total (2,084)—more of an efficiency drop than just replication overhead. The MIX load is again interesting, a 70 percent read/30 percent write split results in a total ops level that's 70 percent of pure read. Most of this drop is again 'pull' from the load incurred by writes to the read part of the test; read I/O drops to 49 percent of pure read instead of 70 percent. On the other hand, writes only dropped IOPS to 88 percent of pure write.

Latency



Average latency at these test loads is mostly sub-10 ms (with the exception of sequential writes at 15 ms). The top two most common latency categories in the sampling ranged from 2/3 to almost 95 percent of the entire latency samples, so the performance is fairly stable as well. Maximum latency ranged from about one second for sequential writes up to almost three seconds for random writes—long but tolerable for block storage error handling.

Configuration guidance

This section covers how to create a Ceph cluster to fit your business needs. The basic strategy of building a cluster is this: with a desired capacity and workload in mind, understand where performance bottlenecks are for the use case, and what failure domains the cluster configuration introduces.

After choosing hardware, Ceph quick start documentation is an excellent place to start for instructions on installing software. Alternately, contact HP and Inktank to help plan and install your Ceph cluster.

Building your own cluster

General configuration recommendations

- The slowest performer is the weakest link for performance in a pool. Typically, OSD hosts should be configured with the same quantity, type, and configuration of storage. There are reasons to violate this guidance (pools limited to specific drives/hosts, federation being more important than performance), but it's a good design principle.
- A minimum size cluster has at least three compute nodes hosting OSDs to distribute the three replicas. A minimum recommended size cluster would have at least six compute nodes. The additional nodes provide more space for unstructured scale, help distribute load per node for operations, and make each component less of a bottleneck.
- If the minimum recommended cluster size sounds large, consider whether Ceph is the right solution. Smaller amounts of storage that don't grow at unstructured data scales could stay on traditional block and file, or leverage an object interface on a file-focused storage target. Smaller Ceph clusters do make sense if the use case requires features of Swift/S3 RESTful interfaces. If the planned solution starts small but scales quickly past the minimum cluster size, then it will benefit from the features of Ceph on HP hardware.
- Ceph clusters can scale to exabyte levels, and you can easily add storage as needed. But failure domain impacts must be considered as hardware is added. Even three-way replication may reach an unacceptable data durability level with enough OSDs. Also, what may have been a sufficient failure domain in the initial CRUSH map may not be a good representation as network and power elements are added. Design assuming elements will fail at scale.

Cluster sizing

Compute and memory

For the OSD hosts, the recommendation is reserving 1 GHz from a core of Intel Xeon processing per OSD daemon. If other tasks run on these cluster nodes, consider the sample data in the CPU results chart under the canned tests as a fairly optimal baseline, and select CPUs resources accordingly. Balance the power of the CPU selected for hardware vs. failure domain considerations for losing the processing power. Even if there are enough free CPU cycles to run VMs or other Linux services on cluster components, more functionality will be lost if a box running multiple services goes down.

From the official Ceph recommendations, monitors should reserve about 1 GB of RAM per daemon instance. The object gateway does not require much buffer for object size load either; in total the sample reference configuration only needed to reserve a few GB on top of other OS and application requirements.

The general memory recommendation is about 2 GB of memory per OSD. Normal I/O usage is rated about 500 MB of RAM per OSD daemon instance; observations haven't shown much of a memory load during normal operation. During recovery however, OSDs may use significantly more memory. The canned tests show extra RAM will have a noticeable positive impact with file system cache on smaller object I/Os, so additional memory can benefit performance too.

Choosing disks

Choose how many drives are needed to meet performance SLAs. That may be the number of drives to meet capacity requirements, but may require more spindles for performance or cluster homogeneity reasons.

Object storage requirements tend to be primarily driven by capacity, so consider required capacity first. Replica count is the biggest impact between raw and real capacities. There will be additional configuration loss factor for things like journal capacity, file system format, and logical volume reserved sectors that will factor into storage efficiency—but these are significantly less impact than replication. A good estimate ratio to use with the sample reference configuration is 1:3.2 for usable to raw storage.

Three-way or greater replica count allows for more distribution of object copies to service reads, but also provides for a quorum on object coherency. Importantly, two disks failing can't cause data loss at these replica levels.

Choose the types of drives to meet requirements—balanced based on price and performance sensitivity—and if SSDs will be used for journal data. Extrapolate from performance results vs. the business use case to help make this selection. HP drive qualification helps maintain homogeneity here, as drives of the same class and capacity are tuned to have similar performance characteristics regardless of vendor. Unstructured data may not require the performance and 24x7 nature of enterprise -class drives. If this is true for the use case, choose drives that trade performance and availability for cost/GB. As an example, HP midline drives are capable of about 550 TB/year of workload and have both SAS and SATA interfaces.

It's a good idea to buffer some performance in estimates. Complex application loads are not as easy to gauge as a simple canned test load, and production systems shouldn't run near the edge so they can better cope with failures and unexpected load spikes.

Some other things to remember around disk performance:

- Replica count means multiple media writes for each object PUT.
- Peak write performance of spinning media without separate journals is around half due to writes to journal and data partitions going to the same device.
- With a single 10 GbE port, the bandwidth bottleneck is at the port rather than controller/drive on any fully disk-populated HP ProLiant SL4540 Gen8 Server node. The controller is optimally capable of about 3 GB/sec, while the effective peak node bandwidth on a 10 GbE link looks to be in the 900 MB–1 GB/sec range out of theoretical 1.25 GB max.
- At smaller object sizes, the bottleneck tends to be on the object gateway's ops/sec capabilities before network or disk. In some cases, the bottleneck can be the client's ability to execute object operations.
- Given the fairly randomly distributed I/O load for object data best case average performance from spinning media is about 90–100 MB/sec. Real world object gateway performance is more in the 60–70 MB/sec average range per disk. This is also impacted by object gateways not providing a particularly deep I/O queue in observed tests. Peak disk performance can be higher, which is why a 4:1 SSD journal ratio is recommended.

Capacity vs. object count

If the use case focuses on many small objects, it may be necessary to get involved in the details of the file systems mounted on each OSD. Because RADOS objects are represented as files, they require an inode to be allocated. Depending on the file system used and the average object size, it may be necessary to change formatting options to maximize disk usage.

As an example, we'll refer to limits for the sample reference configuration. The Ceph-deploy program sets up xfs file systems with 5 percent of capacity as maximum usable for inodes (xfs dynamically allocates inodes as needed). As an example, using 2 KB xfs inode size on 3 TB drives configured as RAID0 results in about 73.2 million inodes available per drive. Clearly these settings would max out inode usage with 1 k objects well before the drive was full of object data.

If inode limitations are a concern, plan file system format parameters before installing Ceph on the cluster. Installation of the OSDs is more involved with custom file system settings; reference the Ceph official documentation for details.

Allocating disks to OSD hosts

Choose the server that fits use case needs; for the OSD hosts we'll cover choices using the HP ProLiant SL4540 Gen8 Server. 3 x 15 units maximize per-node disk utilization on smaller network pipes or offer a greater network bandwidth to disk ratio. Using 3 x 15 HP ProLiant SL4540 Gen8 Servers increases compute density in the rack, but would be the least dense choice for storage. The 2 x 25 and 1 x 60 configurations increasingly improve storage density in the rack at the expense of compute density, and are therefore good choices for progressively 'colder' storage.

Take the drive pool from the first step and divide it into the desired HP ProLiant SL4540 Gen8 Server node configuration. If SSD journals have been chosen, they'll reduce capacity per node accordingly. As an example, an HP ProLiant SL4540 Gen8 Server with a 4:1 ratio of spinning to SSD would have 12 spinning disks per node on a 3 x 15, 20 spinning disks per node on a 2 x 25. SSD journals are not recommended on a 1 x 60 density optimized configuration. Replacing a spinning media slot with SSDs is counter to the focus on density, and the attempt to increase drive write performance runs into server architectural limitations—for example, ratio of disk to network bandwidth.

As part of designing toward homogeneity, adjust drive counts to divide storage into compute evenly where possible. Once the number of disks is chosen, decide how storage will be configured in logical volumes—see Logical Drive Configuration under Cluster Tuning—and select system CPU and memory to match the number of OSDs.

Choosing a network infrastructure
Consider desired bandwidth of storage calculated above, the overhead of replication traffic, and the network configuration of the object gateway's data network (number of ports/total bandwidth). Details of traffic segmentation, load balancer configuration, VLAN setup, or other networking configuration/best practice are very use-case specific and outside the scope of this document.

- Typical choices of configuration for data traffic will be 1–2 1 GbE or 10 GbE networks. Cold object storage use cases may be satisfied with data access over lower bandwidth, but consider that 10 GbE is also useful for faster rebuild and recovery between OSDs. Replicating 1 TB of data across a 1 GbE network takes three hours, with 10 GbE it would be 20 minutes. If more network ports are needed, an additional NIC can be placed in the server's PCIe slot.
- Network redundancy (active/passive configurations, redundant switching) is not recommended, as scale-out configurations gain significant reliability from compute and disk node redundancy and proper failure domain configuration. Consider the network configuration (where the switches and rack interconnects are) in the CRUSH map to define how replicas are distributed.
- A cluster network offloads replication traffic from the data network, and provides an isolated failure domain. With tested replication settings, there are two writes for replication on the cluster network for every actual I/O. That's a significant amount of traffic to isolate from the data network.
- It is recommended to reserve a separate 1 GbE network for management as it supports a different class and purpose of traffic than cluster I/O.

Matching object gateways to traffic
Start by selecting the typical object size and I/O pattern then compare to the sample reference configuration results. The object gateway limits depend on the object traffic, so accurate scaling requires testing and characterization with load representative of the use case. Here are some considerations when determining how many object gateways to select for the cluster:

- Object gateway operation processing tends to limit small object transfer. File system caching for GETs tends to have the biggest performance impact at these small sizes.
- For larger object and cluster sizes, gateway network bandwidth is the typical limiting factor for performance.
- HP has observed around a peak of 3,000–5,000 ops/sec per object gateway testing across object sizes; that range was seen at small object sizes. Maximum practical bandwidth limits seen were in the 900 MB–1 GB/sec range on a 10 GbE link.
- Load balancing does make sense at scale to improve latency, IOPS, and bandwidth. Consider at least three object gateways behind a load balancer architecture.
- Very cold storage or environments with limited clients may only ever need a single gateway.

With the monitor process having relatively lightweight resource requirements, the monitor can run on the same hardware used for an object gateway. Performance and failure domain requirements dictate that not every monitor host is an object gateway, and vice versa. To maximize client traffic per object gateway or meet strictest failure domain requirements, it is recommended the two roles be hosted on separate hardware.

Planning monitor count
Use a minimum of three monitors for a production setup. While it is possible to run with just one monitor, it's not recommended for an enterprise deployment, as larger counts are important for quorum and redundancy. With multiple sites it makes sense to extend the monitor count higher to maintain a quorum with a site down.

Use physical boxes rather than VMs to have separate hardware for failure cases. Do not run a monitor on the same box as OSDs; Ceph documentation recommends avoiding that due to the monitor's usage of fsync() impacting OSD performance.

Cluster tuning

This section contains tuning guidance, which HP considered important to general system configuration.

Placement groups
The tested ratio (for the sum of all pools) recommended by online documentation is

<total_placement_group_count>= ((# OSD * 100) / replica count)

Some tuning heuristics:

- When balancing PG usage for all pools, the proportion of PGs allocated should be based on which pool contains the most objects. So the data pool for the object gateway would typically get the lion's share of the placement groups. If there are multiple pools with high numbers of objects—for example, RBD pools are also created—tuning PG count becomes more complicated.
- Right now pg_num and pgp_num must be the same. Remember to set both values when pools need tuning.
- The *100 ratio can actually vary between about 50–100, where lower counts may help with lower powered systems. For the HP ProLiant SL4540 Gen 8 Server under test, plenty of compute resources are available so a higher number works.
- Powers of two are documented as slightly more performant. It is not practical to jump heavily utilized pools a full power of two every time OSDs are added, but keep this type of growth in mind for planning.
- PG allocations must keep a minimum PG count per OSD for the cluster. Running 'ceph –s' will warn if under threshold.
- PG count in a pool can't be lowered; pools must be deleted and remade to lower PG count (if data isn't important) or copy pool contents to another through RADOS before deletion. So increasing placement groups isn't directly reversible.
- Higher PG counts take more CPU and rebalance time in exchange for better cluster distribution of objects. Changing PG count also incurs a rebalance.

Adding extra PGs for future expansion of OSDs on a critical pool can make sense, or PGs can be left available for RBD pool(s). Best practice depends on current and planned cluster use.

SSD journal usage
If data requires significant PUT performance, consider SSDs for data journaling.

Advantages

- Separation of the highly sequential journal data from object data—which is distributed across the data partition as RADOS objects land in their placement groups—means significantly less seeking to the front of the drive for a journal commit and then seeking elsewhere to write data. It also means that all bandwidth on the spinning media is going to data I/O, approximately doubling bandwidth of PUTs/writes.
- Using an SSD device for the journal keeps storage relatively dense because multiple journals can go to the same higher bandwidth device while not incurring rotating media seek penalties.

Disadvantages

- Each SSD in this configuration is more expensive than a drive that could be put in the slot. Journal SSDs reduce the maximum amount of object storage on the node.
- Tying a separate device to multiple OSDs as a journal and using xfs—the default file system with ceph-deploy—means that loss of the journal device is a loss of all dependent OSDs. With a high enough replica and OSD count this isn't a significant additional risk to data durability, but it does mean architecting with that expectation in mind. The btrfs file system avoids this limitation, but it is not mature enough for some enterprises.
- OSDs can't be hot swapped with separate data and journal devices.

Configuration recommendations

- For bandwidth, four spinning disks to one SSD is a recommended performance ratio. It's possible to go with a higher ratio of spinning to solid state, but that increases the number of OSDs affected by an SSD failure. Also, the SSD could become a bottleneck; larger ratios of disks to SSD journal should be balanced vs. peak spinning media performance.
- Journals don't require a lot of capacity but larger SSDs do provide extra wear leveling. Journaling space reserved by the Ceph should be 10–20 seconds of writes. If each spinning disk peaks at about 150 MB/sec, then 4 GB of capacity in a given journal partition is more than a spinning disk will need to meet those buffer requirements.
- A RAID1 of SSDs is not recommended. Wear leveling makes it likely SSDs will be upgraded at similar times. The doubling of SSDs per node also reduces storage density and increases price per gig. With massive storage scale, it's better to expect drive failure and plan so failure is easily recoverable and tolerable.
- Choose SSDs that match data usage. Consider the number of times the entire device will be written per day vs. the capabilities of the device. If write bandwidth required is in occasional bursts, SLC flash doesn't make sense.

Logical drive configuration

For a 1 x 60, significant CPU cycles must be reserved for 60 OSDs on a single compute node. A 1 x 60 HP ProLiant SL4540 Gen8 Server fully-loaded could reduce CPU usage by configuring RAID0 volumes across two drives at a time—resulting in 30 OSDs. Configuring multiple drives in a RAID array can reduce CPU cost for colder storage, in exchange for reduced storage efficiency to provide reliability. It can also provide more CPU headroom for error handling, or additional resources if cluster design dictates CPU resource usage outside of cluster specific tasks.

Bill of materials

This BOM reproduces the sample reference configuration.

Note

HP ProLiant servers ship with an IEC-IEC power cord for rack mounting.

HP ProLiant SL4540 Gen8 Server

Quantity	Part number	Description
5	663600-B22	HP ProLiant SL454x 2X Node Chassis
10	664644-B22	HP 2 x SL4540 Gen8 Tray Node Server
10	684373-L21	HP SL4540 Gen8 Intel Xeon E5-2470 (2.3 GHz/eight core/20 MB/95 W) FIO Processor Kit
60	647897-B21	HP 8 GB (1 x 8 GB) Dual Rank x4 PC3L-10600R (DDR3-1333) Registered CAS-9 Low Voltage Memory Kit
10	664648-B21	HP SL4500 10 GbE I/O Module Kit
10	655874-B21	HP QSFP/SFP+ Adaptor Kit
10	692276-B21	HP Smart Array P420i Mezz Controller FIO Kit
10	631679-B21	HP 1 GB FBWC for P-Series Smart Array
10	668943-B21	HP 12 in Super Cap for Smart Array
10	655708-B21	HP 500 GB 6G SATA 7.2 k 2.5 in SC MDL HDD
200	652766-B21	HP 3 TB 6G SAS 7.2 k 3.5 in MDL SC HDD
50	691864-B21	HP 200 GB 6G SATA 2.5 in SC Enterprise SSD
10	656364-B21	HP 1,200 W CS Platinum Power Supply kit
5	681254-B21	HP 4.3U Rail Kit
0	681260-B21	HP 0.66U Spacer Blank Kit—Available, but not used for this configuration.
10	512485-B21	HP iLO Adv 1 Server including one-year TS&U software

HP ProLiant DL360p Gen8 Server

Quantity	Part number	Description
3	654081-B21	HP ProLiant DL360p Gen8 8 SFF Server
3	654772-L21	HP DL360p Gen8 E5-2650 FIO Kit
3	654772-B21	HP DL360p Gen8 E5-2650 Kit
24	690802-B21	HP 8 GB 2R x 4 PC3-12800R-11 Kit
3	684208-B21	HP Ethernet 1 GbE 4P 331FLR FIO Adapter
3	665249-B21	HP Ethernet 10 GbE 2P 560SFP+ Adapter
6	652572-B21	HP 450 GB 6G SAS 10 k rpm SFF (2.5-inch) SC Enterprise three-year Warranty Hard Drive
3	631679-B21	HP 1 GB FBWC for P-Series Smart Array
3	339778-B21	HP Raid1 Drive 1 FIO Setting
6	503296-B21	HP 460 W CS Gold Hot Plug Power Supply Kit
3	663200-B21	HP 1U FIO Friction Rail Kit
3	512485-B21	HP iLO Adv 1 Server including one-year TS&U software

HP Networking cables

Quantity	Part number	Description
20	263474-B23	HP IP CAT5 Quantity eight 12 ft/3.7 m cable
6	263474-B22	HP IP CAT5 Quantity eight 6 ft/2 m cable
3	JD096C	HP X240 10G SFP+ to SFP+ 1.2 m DAC cable
20	JD097C	HP X240 10G SFP+ to SFP+ 3 m DAC cable
2	JG328A	HP X240 40G QSFP+ QSFP+ 5 m DAC cable

HP 1 GbE switch

Quantity	Part number	Description
1	J9728A	HP 2920-48G switch, 1 J9739A Power Supply Included
1	U6319E	Three-year Support Plus, four-hour onsite, 24x7 coverage
1	U4830E	HP Networks Stackable Legacy Switch Startup Service
1	U4826E	HP Networks Stackable Legacy Switch Installation Service

HP 10 GbE switches

Quantity	Part number	Description
2	JC772A	HP 5900AF-48XG-4QSFP+ Switch
4	JC680A	HP 58x0AF 650W AC Power Supply
4	JC682A	HP 58x0AF Back (power)-Front (ports) Fan Tray
2	U5Y06E	HP three-year SupportPlus24 5900-48 swt Service [for JC772A]

HP Rack and Power

Quantity	Part number	Description
1	BW908A	HP 642 1,200 mm Shock Intelligent Rack
1	BW932A	HP 600 mm Rack Stabilizer Kit
1	BW930A	HP Air Flow Optimization Kit
1	BW909A	HP 42U 1,200 mm Side Panel Kit
2	AF916A	HP 3PH 48A NA/JP Power Monitoring PDU
2	AF500A	HP 2, 7X C-13 Stk Intl Modular PDU
1	120672-B21	HP 9000 Series Ballast Option Kit

Summary

With rapid growth of unstructured data and backup/archival storage, traditional storage solutions are lacking in their ability to scale or efficiently serve this data. The cost per gigabyte for SAN and NAS at scale is undesirable, and the solutions provide performance features data doesn't really require. Tape has better cost at scale, but doesn't always meet latency requirements for data access. Management of the quantity of storage and sites is complicated; guaranteeing enterprise reliability to the clients becomes difficult or impossible.

HP and Ceph on Linux uses object storage and industry-standard servers to provide the cost, reliability, and centralized management businesses need for petabyte unstructured storage scale and beyond. Industry-standard server hardware from HP is a reliable, easy-to-manage, and supported hardware infrastructure for the cluster. Ceph and Inktank provide the same set of qualities on the software side. Together, they form a solution with a lower TCO than traditional storage that can be designed and scaled for current and future unstructured data needs.

Importantly, the solution brings the control and cost benefits of open source to those enterprises that can leverage it. Open source software doesn't require additional license costs. There's no inherent vendor lock-in from the cluster software. Source code is available to control and customize what's deployed in the data center. Ceph can also be a key backing component of OpenStack Cinder and Glance.

Software, storage, and network infrastructure can be scaled to solve your exploding data problems. Ceph cluster software and HP hardware are a compelling solution to a new scale of storage requirements, freeing your storage from traditional limitations.

Implementing a proof-of-concept

As a matter of best practice for all deployments, HP recommends implementing a proof-of-concept using a test environment that matches the planned production environment as closely as possible. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof-of-concept, contact an HP Services representative (hp.com/large/contact/enterprise/index.html) or your HP partner.

Glossary

- Cold, warm, and hot storage—Temperature in data management refers to frequency and performance of data access in storage. Cold storage is rarely accessed and can be stored on the slowest tier of storage. As the storage 'heat' increases, the bandwidth over time, as well as instantaneous (latency, IOPS) performance requirements increase.
- CRUSH—Controlled Replication Under Scalable Hashing. The algorithm Ceph uses to compute object storage locations.
- Epoch—Ceph maintains a history of each state change in the Ceph Monitors, Ceph OSD Daemons, and PGs. Each version of cluster element state is called an "epoch."
- Failure domain—An area of the solution impacted when a key device or service experiences failure.
- Federated storage—A collection of autonomous storage resources with centralized management that provides rules about how data is stored, managed, and moved through the cluster. Multiple storage systems are combined and managed as a single storage pool.
- Object storage—A storage model focusing on data objects instead of file systems or disk blocks; objects have key/value pairs of metadata associated with them to give the data context. Typically accessed by a REST API, designed for massive scale, and using a wide, flat namespace.
- PGs—Placement Group. A grouping of objects on an OSD; pools contain a number of PGs and many PGs can map to an OSD.
- Pools—Logical partitions for storing objects. Pools set ownership/access to objects, the number of object replicas, the number of placement groups, and the CRUSH rule set to use.
- RADOS—A Reliable, Autonomic Distributed Object Store. This is the core set of storage software that stores the user's data in a Ceph Cluster (MON+OSD).
- REST—Representational State Transfer is stateless, cacheable, layered client-server architecture with a uniform interface. In this cluster, the REST APIs are architected on top of HTTP.

For more information

With increased density, efficiency, serviceability, and flexibility, the HP ProLiant SL4540 Gen8 Server is the perfect solution for scale-out storage needs. To learn more visit: hp.com/servers/sl4540.

To support the management and access features of object storage, and seamlessly operate as part of HP Converged Infrastructure, the HP ProLiant DL360p Gen8 series brings the power, density, and performance required.

HP OneView helps companies of all sizes unlock the value of converged infrastructure by bringing the best of consumer IT to the data center and allowing teams to work in a more natural and collaborative way. Visit: hp.com/go/oneview.

HP Integrated Lights-Out simplifies server setup, promotes remote administration, engages health monitoring, and maintains power and thermal control. For more information see: hp.com/go/ilo.

HP simplifies, integrates, and automates networking so organizations can focus on what they do best. Visit hp.com/go/networking for more information. The HP switches used in this document are HP 2920-48G and HP 5900AF-48XG-4QSFP+-48G.

Documents for HP scale-out object storage solutions on industry-standard servers are at hp.com/go/objectstorage.

Ceph has excellent documentation available at its website; this white paper has sourced it extensively. The documentation master page starts here: ceph.com/docs/master/.

To help us improve our documents, please provide feedback at hp.com/solutions/feedback.

4AA5-2799ENW, June 2014