INTEL® SOLUTIONS REFERENCE ARCHITECTURE

Analyzing Ceph Cluster I/O Performance to Optimize Storage Costs

Datagres PerfAccel™ Solutions with Intel® SSDs

DATAGRES
Application I/O Intelligence

"PerfAccel uses rich and indepth I/O analytics to determine the right SSD size, and provides enhanced performance using intelligent placement of data in the cache."

Audience & Purpose

For companies, CIOs, IT Managers, System Administrators and any user having a Virtual Machine environment (like KVM* or cloud deployment based on OpenStack*) on top of a Ceph* cluster. The purpose of this reference architecture is to show how PerfAccel™ can be used in a Ceph cluster to get better and deeper insights/analytics about the VMs. When used with Intel® SSDs the PerfAccel cache provides accelerated I/O, resulting in faster VMs and lower cost of storage. PerfAccel analytics provides information about data distribution (among jobs, nodes), access patterns and bottlenecks, in addition to performance metrics. PerfAccel accelerates application performance through dynamic data placement & management on Intel SSDs. The performance improvements translate to better VM performance and scaling allowing the same resources can be used to host more VMs, translating into a more optimized and cost effective datacentre. This reference architecture focuses on an example deployment demonstrating how PerfAccel provides better performance and a richer set of analytical data for a Ceph cluster hosting OpenStack virtual machine disk images.

Table of Contents

Executive Summary

Ceph is a distributed storage system providing object, block and file storage that can be "like nirvana for shared storage"[1] across distributed COTS server hardware. It is also supported in the OpenStack project, so virtual machine (VM) disk images can be directly hosted on Ceph as RADOS block devices.

PerfAccel can be treated as a Data Service in the Software Defined Storage (SDS) Layer, which does the dynamic data placement of hot data on high performance SSDs and also provides a rich analytical framework for analyzing workloads in a seamless and fine grained manner. This Solutions Reference Architecture (SRA) contains details about the data placement aspects as well as the rich analytics aspect of the Datagres PerfAccel product.

PerfAccel can cater to a wide range of storage usage models. Common usage models include business database (OLTP, OLAP), application data (webserver, email, VM boot), business intelligence data (Hadoop, Large relational databases), Distributed storage (Ceph) and a lot more. In this SRA, we will focus on the beneficial features of PerfAccel in a Ceph cluster in one of its popular usage models: providing the virtual block devices to VMs in an OpenStack project.

The paper demonstrates the value of Intel SSDs used along with PerfAccel on a Ceph cluster, using Intel Xeon processors. PerfAccel is able to use the consistent high performance and low latency behavior of Intel SSDs to show large improvements in the range of 5x to 8x for Sequential and Random I/O workloads. The high performance of Intel Xeon processor ensures there is no noticeable overhead, even while PerfAccel is able to extract in-depth information from the storage layer. PerfAccel when used without SSDs, can still provide all the in-depth analytics for Ceph.

Such usage can help determine the I/O footprint of the cluster, and provide concrete data on the need and size of SSDs for better performance.

Enterprise storage requirements tend to grow at least as fast as the underlying business does. A Ceph based shared storage system can 'scale-out' to meet demand by adding hardware to the SDS system. A shared storage Ceph system built with PerfAccel data service and Intel based servers and SSDs will elevate the immediate and long term storage system ROI through the tools to add scale-out capacity without growing un-needed complexity.

Introduction: PerfAccel & Issues It Solves In A Ceph Cluster

Why Choose Ceph ?

In this age of virtualization, it's very common to see KVM servers or OpenStack servers hosting a bunch of virtual machines (VMs) with their disks coming from either local storage or remote storage like NFS/iSCSI etc.

The performance of these VMs directly depends on their attached virtual disks. If the virtual disks are being served from multiple physical disks in the same system, then the total I/O bandwidth is limited to the combined bandwidth of all the physical disks. If the virtual disks are being served from network storage, then the bandwidth is again limited by things like storage server's performance, how many clients are using the storage server, available network bandwidth, etc.

Ceph can be used in the above scenario to improve the performance of the VMs. Ceph combines the I/O bandwidth of multiple storage systems, designated as Ceph storage nodes, and provides it to the connected clients i.e. the KVM/OpenStack servers. And new Ceph nodes can be added as and when needed to get more I/O bandwidth. Ceph improves the I/O performance by

parallelizing it across different storage nodes. Ceph is based on remote block devices (RADOS), which can be directly attached to VMs as network disks. Ceph OSD or the Object Storage Device, is a physical or logical storage unit. It is also used to refer to the Ceph OSD Daemon.

- Ceph provides higher performance by combining I/O bandwidth of multiple storage nodes
- Max Performance in Ceph cluster gets limited because of Spinning disks
- Absence of a Rich analytical framework for Ceph Clusters makes it difficult to monitor I/O patterns & behavior of VMs

Performance Limitations Of Ceph?

In general, Ceph Clusters are built on top of inexpensive SATA Hard disk drives. Though the ultimate aim of Ceph is to provide higher I/O performance by distributing the I/O in parallel across multiple storage nodes, there will still be a hard limit to the performance due to the mechanical constraints of the spinning SATA disks. To overcome this problem, one solution is to replace all the SATA disks with high performance SSDs, but that will increase the cost signifigantly when applied to the whole cluster. To get a balance between cost/performance, we need to understand the access frequency and put only the data which is being accessed really frequently on SSD. So, what if it was possible to dynamically place the hot data on the SSDs and perform the I/O from/to the SSDs, without any manual intervention? This is where PerfAccel comes into the scenario. PerfAccel does exactly that i.e. intelligent and dynamic data placement. PerfAccel supports acceleration of all I/O across multiple platforms like NAS, SAN and DAS to provide a seamless performance benefit to all types of applications. Configurable caching policies ensure that the right working set resides in the cache for maximum performance benefit.

About PerfAccel

PerfAccel presents a unique solution that provides deep analytics to observe I/O behavior, helping determine better data placement and improve performance. In addition, using its intelligent caching capabilities, PerfAccel can deliver much higher performance. The result is a significant reduction in infrastructure costs while providing rich analytics and much higher performance.

PerfAccel supports acceleration of all I/O across multiple platforms. Configurable caching policies ensure that the right working set resides in the cache for maximum performance benefit. It is extremely easy to deploy and manage, and the in-depth analytics can provide deep insights to help users understand application I/O pattern and I/O footprint to optimize workloads.

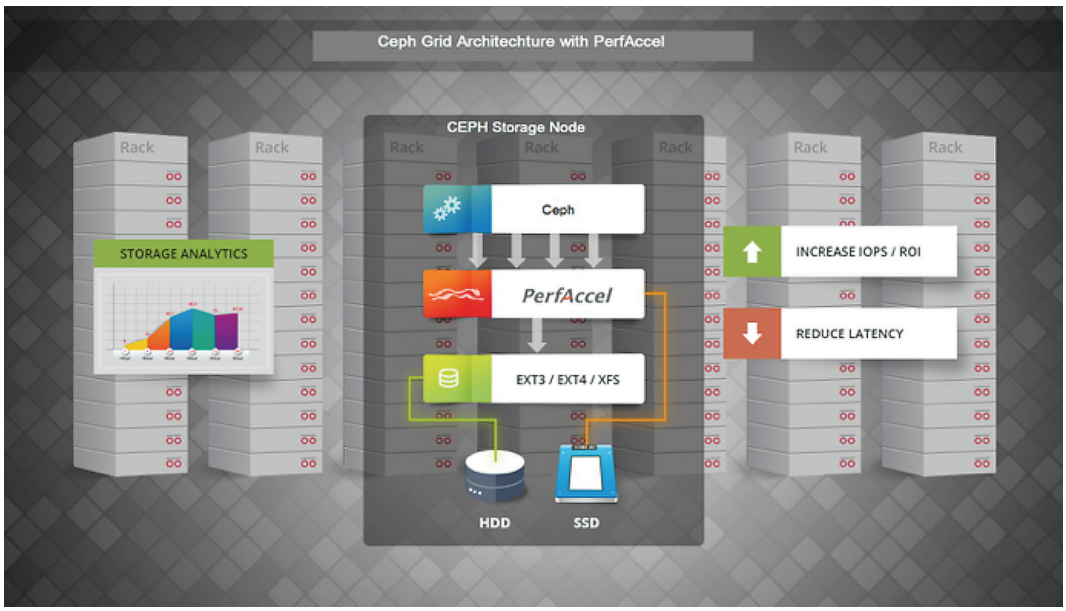


Figure 1 - Ceph Grid Architecture with PerfAccel

PerfAccel can be used to extract in-depth analytics of workloads and their I/O patterns. The deep analytics helps PerfAccel to then intelligently determine the benefits that can be achieved if a SSD cache is used. While helping users gain deeper insights regarding their workload and I/O behavior, PerfAccel analytics can also help determine the size of SSD cache required and the associated benefits.

PerfAccel can be used to take advantage of SSDs in a manner which is beneficial, even while data resides on conventional devices, by ensuring the hot data resides on the faster storage thereby providing much better performance. PerfAccel would use the faster device available as a cache and will ensure optimal placement of frequently used hot data. In addition, since the majority of read operations are offloaded to the cache, the back end storage device which holds the entire dataset is more responsive as it has to serve fewer IOPS. Thus PerfAccel cache not only improves read performance, it also implicitly improves the write performance of the application.

Analytical Needs In A Ceph Cluster?

In any Ceph cluster, which hosts the disk images of virtual machines as RADOS block devices, there is always a need to know about the performance, behavior and access patterns of each VM. A single VM can hog a larger chunk of the I/O bandwidth of the Ceph cluster, leading to resource starvation of the disks for all other VMs. The end effect is that the applications hosted on those VMs performs poorly. The sysadmin then has to go through a tedious process and a large set of information to isolate and pin point the problem to the particular resource hogging VM. Ceph does not have any specific tool for this.

The rich analytical framework of

Perfccel Provides:

- Storage visibility through deep file-level analytics
- Intelligent caching & deterministic placement of hot data
- Higher performance from each SSD used optimally

PerfAccel provides a wide range of metrics, in a fine grained manner, which can be utilized to detect and pinpoint such issues. The Ceph customized analytics from PerfAccel can throw light on various aspects starting from the individual virtual disk level to the storage node level in the Ceph cluster. We can logically categorize the analytical data made available by PerfAccel into the following:

Device Level

This category of analytical data will provide a brief snapshot of what's going on with all the VM disks that are present in the Ceph cluster (in all storage nodes). The data will be available for every single disk and would comprise of metrics like:

a. Percentage and size of the disk which is really accessed and cached

b. Amount of cached reads and writes done on each disk

With the help of these metrics, users can easily figure out which disk is really being used, their usage details, and for what purpose (read or write). Then we can drill-down further to get more detailed analytics data for any particular disk of our choice.

RBD Level

This category of analytical information is oriented towards individual disks of the VMs. Analytical data can be displayed for disks in a selective manner and would contain metrics like:

a. Object level distribution info for the disk, across nodes

b. Information about cached Ceph objects of the disk, across nodes

c. Cache utilization info of Ceph objects across Nodes

d. Information regarding hot areas of the disk, both in terms of offset as well as objects

e. Very detailed and drilled-down metrics for each object of the disk, in each node.

This information will be displayed for every Ceph Node where the disk has objects.

With these kind of insights, users can get answers to questions like how many reads/writes has happened from the cache for a particular Ceph object, when was it last accessed, its resident time in cache etc.

OSD Level

This category of analytics provides data about the I/O patterns and behavior of each OSD, in each node. Some of the important metrics available here are:

a. I/O metrics, like how much Read, Write, Attribute Read and Write, has happened on the OSD, for both cached and un-cached access

b. IOPS metrics (Bandwidth, Latency, Throughput), for both cached and un-cached I/O

c. Cache utilization metrics, including number of objects cached.

With this information, we can easily find out if any of the OSDs are under-

performing or hogging all of the disk bandwidth resulting in slowing down of other OSDs, or how the I/O is distributed across all OSDs i.e. uniformly or in a random manner, I/O latencies and other such aspects.

Node level

This category of information sheds light on the I/O behavior of the Ceph storage nodes.

Various data points available under this are :

a. Number of Ceph objects cached in the node

b. Cache utilization on the node

c. Amount of cached reads and writes done on the Node

With this kind of data, we can gain some useful insights into the Ceph cluster, like if the nodes are balanced properly or not, whether any nodes are under performing.

We can see PerfAccel analytics and caching information at any point of time in two ways :

a. In any Ceph node, by executing the various PerfAccel commands (`dgpctl`)

b. In the management node, by using PerfAccel Ceph Analyzer

Testbed Architecture & Configuration

This below figure shows the Testbed setup. For this reference architecture, Ceph is deployed in 4 storage nodes. And there are two OpenStack hosts, where the virtual machines are running and generating I/O.

All the 4 storage nodes and the OpenStack hosts are connected to 10GbE network. The storage nodes are installed with Red Hat Enterprise Linux (RHEL)* 6.5 OS and the OpenStack hosts are installed with Ubuntu* 12.04.

The client VMs hosted on the OpenStack hosts are all Ubuntu 12.04. For load generation, fio* is used inside all the VMs. Fio is short for flexible I/O, a freely available and widely used workload generation tool.

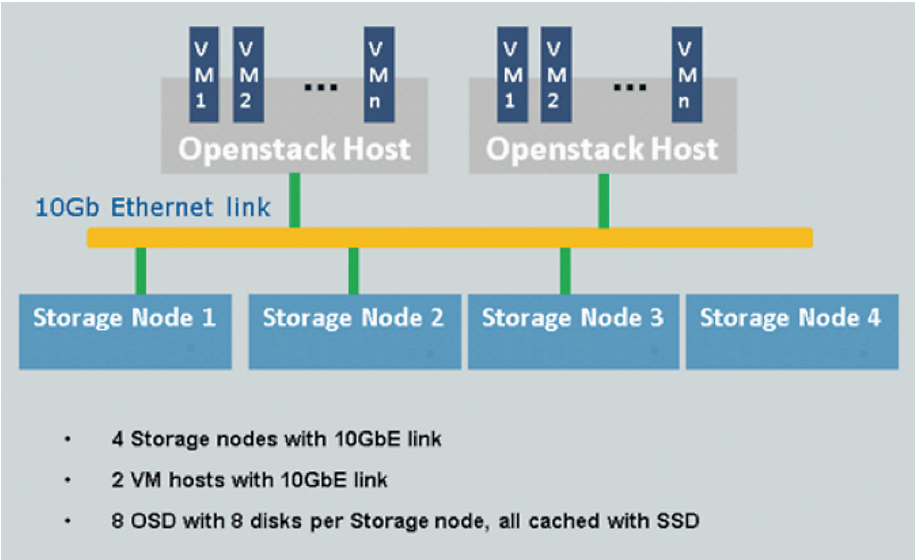


Figure 2 - Testbed Setup Architecture

System Configurations

CEPH Storage Node	
CPU	1 x Intel® Atom™ CPU C2750 8C @ 2.40GHz
Memory	8 GB
NIC	1 x Intel® X540-AT2 10GbE
SATA Controller	2 x Marvell* 88SE9235
Disks	8 x Seagate* 3.5" 3TB 7200 RPM SATA HDD (Data) 1 x Seagate 3.5" 3TB 7200 RPM SATA HDD (System) 2 x Intel® 2.5" 480G DC3500 SSD (Cache & Journal)
OS	CentOS* 6.5
Kernel	2.6.32-431
Ceph	0.72.2
FS	XFS

OpenStack Host	
CPU	Intel® Xeon® CPU E5-2640 6C @ 2.50GHz
Memory	64 GB
NIC	1 x 10GbE
Disks	1 x Seagate* 3.5" 500G 7200 RPM SATA HDD (System)
OS	Ubuntu* 12.04.3 LTS
Kernel	3.8.0-29
Guest VM	
CPU	1 CPU
Memory	2 GB
Disks	1 x System Disk (Local) 1x 60GB Ceph RADOS Disk
OS	Ubuntu 12.04.3 LTS
Kernel	3.8.0-29

Ceph Cluster

Ceph 0.72.2 version was installed on all four of the storage nodes.

Each storage node was then configured in a similar way:

a. 8 OSDs were created from the 8 x 1TB SATA HDD

b. In each SSD, 4 x 10GB partitions were created

c. 4 Partitions from each SSD served as the Journal for 4 OSDs

d. The OSDs are configured for XFS filesystem

So, at the end we have 32 OSDs in the whole Ceph Cluster.

Disk Configuration In Ceph Storage Node

The figure to the right shows the disk configuration for a Ceph Storage Node.

- Here, all of the 8 OSD data disks are 3TB SATA drives. The OSD journals are on SSDs.
- In each SSD, the first partition was a 400GB one and was reserved for PerfAccel cache.
- Then, 4 more partitions, each of 10GB, were created in each SSD
- Each OSD was configured to use one of the 8 SSD partitions as its journal

Once the OSDs are created, they are mounted as directories under /var/lib/ceph/osd

```
Ex: In Node 1, we have 8 OSDs mounted as:

/var/lib/ceph/osd/ceph-0
/var/lib/ceph/osd/ceph-1
/var/lib/ceph/osd/ceph-2
/var/lib/ceph/osd/ceph-3
/var/lib/ceph/osd/ceph-4
/var/lib/ceph/osd/ceph-5
/var/lib/ceph/osd/ceph-6

/var/lib/ceph/osd/ceph-7
```

Figure 3 - Disk Configuration For Osds In Storage Node

Ceph Configuration

The following configuration was used for Ceph

```
PGs number: 2048
Kernel Parameter: read_ahead_kb=2048

OSD section in ceph.conf:
      osd mount options xfs = rw,noatime,inode64,logbsize=256k
      filestore max inline xattr size = 254
      osd_op_threads = 20
      filestore_queue_max_ops = 500
      filestore_queue_committing_max_ops=5000
      journal_max_write_entries = 1000
      objecter_inflight_ops = 10240
      filestore_queue_max_bytes = 1048576000
      filestore_queue_committing_max_bytes = 1048576000
      journal_max_write_bytes = 1048576000
      journal_queue_max_bytes = 1048576000
      ms_dispatch_throttle_bytes = 1048576000
      objecter_inflight_op_bytes = 1048576000
      filestore_max_sync_interval = 10
      filestore_flusher = false
      filestore_flush_min = 0
      filestore_sync_flush = true

IO Scheduler for Data HDDs: deadline
IO Scheduler for Cache SSDs: CFW (Ubuntu default)
```

Figure 4 - Ceph Configuration Used For Test

PerfAccel Deployment In A Ceph Cluster

Deploying PerfAccel with Ceph is a six step process. The PerfAccel getting started guide may be of help for first time users. The steps are outlined below.

a. PerfAccel was installed using the binary corresponding to the RHEL 6.5 OS and activated by using the necessary license. This was done on all 4 storage nodes.

b. A management node (CentOS 6.5) for PerfAccel was also created and the binary installed there as well. The management node hosts the GUI for PerfAccel and can be used to manage other nodes and also view the analytics.

This node was given password-less access to all the 4 Ceph storage nodes and the 2 OpenStack nodes.

c. Before proceeding further, we stopped the OSD services in all the storage nodes service ceph-osd stop

d. Then, PerfAccel caches were created on the first partitions of the 2 SSDs, in each node.

```
dgpctl –cache create cache1
 /dev/sde1

dgpctl –cache create cache2
 /dev/sdi1
```

e. Now, the OSD mount points were accelerated by creating PerfAccel Sources on each of them

Ex: Let us create it for Node 1, 1st OSD ceph-0

```
dgpctl –source create ceph0
/var/              lib/ceph/
osd/ceph-0 cache1 95G
```

This command created a Source for the first OSD i.e. Ceph-0.

The name of the source is “ceph0” (it can be anything, but to maintain uniformity we named all the sources along the lines of their OSD name)

From the PerfAccel cache, 95GB of cache space was allocated per OSD.

At the end, each PerfAccel cache on each node accommodated 4 OSDs

f. Now, the Ceph OSD services were started again

```
service ceph-osd start
```

The above steps were performed on all of the nodes.

To verify that all the ceph OSDs are up and running, we used the command:

```
ceph osd tree
```

At this point PerfAccel is active on all the nodes and OSDs.

PerfAccel Ceph Analyzer

Using the PerfAccel Ceph Analyzer, a Ceph customized analytics tool, needs a central system which has ssh access to all the Ceph storage nodes as well as the OpenStack host nodes, in a password-less manner.

We had already configured the management node for this purpose.

Before using it, we had to fill in some configuration files, which will be used by the analyzer later on. This is a one-time effort and not required to be done again unless new storage nodes or OpenStack nodes are added.

1. /etc/dgres/config/node.list
   This file contains the ip address or hostname of the Ceph nodes. There are four nodes, and all of them were added to the file.
2. /etc/dgres/config/kvm_host.list
   This file contains the ip address or hostname of the OpenStack (KVM) Host. As we have two such hosts, so their hostnames were added to this file.

The analyzer has two phases :

1. **Collect Phase** – where the analytics data is collected from all of the nodes. It can be run at any point of time by the command :

   dgpctl –ceph collect

2. **Analyze Phase** – This analyzes the collected data and generates various reports based on what the user wants

   ```
   dgpctl –ceph analyze data_
   dir_path [filter]
   ```

Workload Generation & Execution

The I/O workload was generated from 20 virtual machines running on the OpenStack (KVM) hosts (10 VMs in each host). These VMs generated loads using the “fio” benchmarking tool

Every virtual machine has a local OS Disk (/dev/vda) and a remote RADOS block device (/dev/vdb) coming from the Ceph cluster. And fio used the /dev/vdb directly for its I/O.

Fio was executed in “direct io” mode to bypass local memory caching.

Fio was configured with the following workloads :

a. Sequential Read
   Queue Depth: 64, Block Size: 64K

b. Sequential Write
   Queue Depth: 64, Block Size: 64K

c. Random Read
   Queue Depth: 8, Block Size: 4K

d. Random Write
   Queue Depth: 8, Block Size: 4K

And the main goals of the above tests were to observe:

a. Performance boost achieved due to the intelligent & dynamic data placement done by PerfAccel

b. Rich Analytics data for Ceph, provided by PerfAccel

The tests were run 3 times.

1. First run was made on the existing Ceph Cluster i.e. without PerfAccel
2. Second and third runs were made after PerfAccel was installed and configured for use.

In the second run with PerfAccel, the hot data was cached on the SSDs. Therefore the third run with PerfAccel was the one where we expected the performance boost to be visible.

Test Results

We have categorized the performance results in two graphs.

The first graph is for Sequential I/O and the second one is for Random I/O.

In each graph, we have displayed 2 groups of bars. The first group is for Reads and second one is for Writes.

Note: PerfAccel (cached) here refers to the last test run, which means PerfAccel was repeated.

In the graph below (Figure 5), the first group of bars is for Sequential Reads, with a Block size of 64K and queue depth of 64. The second group is for Sequential Writes, with a Block size of 64K and queue depth of 64.

As we had used two OpenStack hosts connected to 10GbE link, the maximum network bandwidth we can get is 20Gbps i.e. close to 2500 MB/s.

We can clearly see that for Sequential Reads, the performance increased by ~ 5x.

There was potential for more increase, but it was limited because of the 20Gbps Network bandwidth. So, clearly we achieved the network line-rate speed with PerfAccel.

PerfAccel was used in Writethrough (Read caching) mode, so we expected the write performance to be approximately the same as without PerfAccel. And the same is visible from the graph.

In the next graph (Figure 6), the first group of bars is for Random Reads, with a Block size of 4K and queue depth of 8. The second group is for Random Writes, with a Block size of 4K and queue depth of 8.

Here, we can see that the 3rd run with PerfAccel has increased the performance of Random reads by ~ 8x. And the write performance is again similar to the case without PerfAccel.

"PerfAccel is able to use the consistent high performance and low latency behavior of Intel SSDs to show large improvements in the range of 5x to 8x for Sequential and Random I/O workloads."

1. Sequential I/O

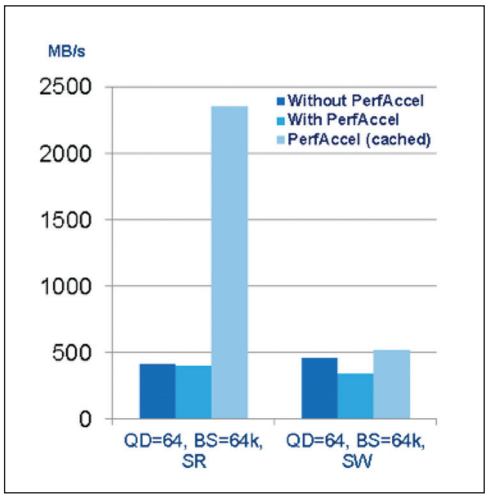


Figure 5 - Sequential Read/Write I/O results in MB/s

2. Random I/O

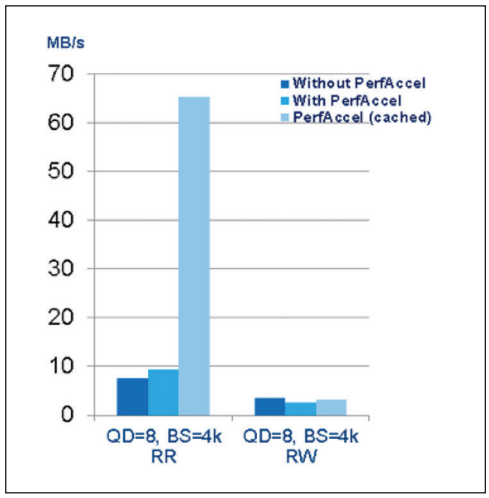


Figure 6 - Random Read/Write I/O results in MB/s

PerfAccel Rich Analytics

While the test was running, we executed various PerfAccel analytics commands on the storage nodes to see how they were doing.

We also generated various reports on the Ceph Cluster and VM disks through the collect and analyze phases of PerfAccel Ceph Analyzer in the management node.

Note: Here, PerfAccel "Source" refers to the un-cached data i.e. The data served directly from the OSDs and "Cache" refers to the cached data i.e. the data served from the PerfAccel caches.

IO Trace (OSD level)

Cache Read	Cache Write	Source Read	Source Write	Source Name	Source Dir
47.3 GB	16.0 KB	4.0 KB	16.0 KB	ceph0	/var/lib/ceph/osd/ceph-0
33.4 GB	12.0 KB	260.0 KB	20.0 KB	ceph6	/var/lib/ceph/osd/ceph-6
30.5 GB	16.0 KB	128.0 KB	16.0 KB	ceph3	/var/lib/ceph/osd/ceph-3
29.2 GB	0.0 KB	0.0 KB	0.0 KB	ceph7	/var/lib/ceph/osd/ceph-7
26.0 GB	8.0 KB	128.0 KB	16.0 KB	ceph1	/var/lib/ceph/osd/ceph-1
23.8 GB	28.0 KB	384.0 KB	28.0 KB	ceph2	/var/lib/ceph/osd/ceph-2
17.2 GB	0.0 KB	0.0 KB	0.0 KB	ceph4	/var/lib/ceph/osd/ceph-4
15.0 GB	4.0 KB	4.0 KB	8.0 KB	ceph5	/var/lib/ceph/osd/ceph-5

Figure 7 - PerfAccel Io Trace Of A Storage Node

This beautifully shows how much of the data was served from the cache and source. The higher the numbers are in Cache Read, the better it is, as they are served from the Intel SSDs.

Cache Capacity	Cache usage	Use%	Max usage	Files in Cache	Source Name	Source Dir
95.0 GB	15.5 GB	16.3%	15.5 GB	3925	ceph4	/var/lib/ceph/osd/ceph-4
95.0 GB	23.4 GB	24.6%	23.4 GB	5922	ceph1	/var/lib/ceph/osd/ceph-1
95.0 GB	27.3 GB	28.7%	27.3 GB	6900	ceph3	/var/lib/ceph/osd/ceph-3
95.0 GB	42.4 GB	44.7%	42.4 GB	10740	ceph0	/var/lib/ceph/osd/ceph-0
95.0 GB	26.2 GB	27.6%	26.2 GB	6649	ceph7	/var/lib/ceph/osd/ceph-7
95.0 GB	30.2 GB	31.8%	30.2 GB	7630	ceph6	/var/lib/ceph/osd/ceph-6
95.0 GB	13.6 GB	14.3%	13.6 GB	3441	ceph5	/var/lib/ceph/osd/ceph-5
95.0 GB	21.4 GB	22.5%	21.4 GB	5408	ceph2	/var/lib/ceph/osd/ceph-2

Figure 8 - PerfAccel Usage Trace Of A Storage Node

Usage Trace (OSD Level

This information gives a brief view of the cache usage by each OSD and the number of Ceph objects cached so far. The values reflect the amount of actual hot data being accessed. It also gives us a view of how the Ceph objects are distributed across OSDs. Here “Files in Cache” are the Ceph Objects.

IOPS Trace (OSD Level)

device type	total read ops	total write ops	total read data (GB)	total write data (GB)	read throughput (MB/s)	write throughput (MB/s)	avg.read latency (ms)	avg.write latency (ms)	Source Dir
Cache	258141	6	31.801	0.000	126.6	18.4	0.997	0.106	/var/lib/ceph/osd/ceph-8
Source	0	6	0.000	0.000	0	3.2	0.000	0.604	/var/lib/ceph/osd/ceph-8
Cache	295568	2	36.459	0.000	136.6	0.1	0.924	0.135	/var/lib/ceph/osd/ceph-13
Source	3	6	0.000	0.000	7.1	3.6	6.232	0.539	/var/lib/ceph/osd/ceph-13
Cache	253667	1	31.316	0.000	125.5	54.3	1.007	0.144	/var/lib/ceph/osd/ceph-14
Source	4	6	0.000	0.000	5.4	1.6	6.133	1.593	/var/lib/ceph/osd/ceph-14
Cache	291316	1	35.939	0.000	135.9	22.9	0.929	0.171	/var/lib/ceph/osd/ceph-15
Source	0	2	0.000	0.000	0	4.4	0.000	0.445	/var/lib/ceph/osd/ceph-15
Cache	180214	4	22.253	0.000	112.9	29.6	1.120	0.132	/var/lib/ceph/osd/ceph-9
Source	19	8	0.000	0.000	3.5	2.8	2.009	0.690	/var/lib/ceph/osd/ceph-9
Cache	261214	0	32.246	0.000	133.4	0	0.947	0.000	/var/lib/ceph/osd/ceph-10
Source	0	0	0.000	0.000	0	0	0.000	0.000	/var/lib/ceph/osd/ceph-10
Cache	324509	2	39.984	0.000	151.5	44.7	0.833	0.087	/var/lib/ceph/osd/ceph-11
Source	3	4	0.000	0.000	11.8	3.4	3.748	0.575	/var/lib/ceph/osd/ceph-11
Cache	246358	0	30.401	0.000	129.9	0	0.973	0.000	/var/lib/ceph/osd/ceph-12
Source	0	0	0.000	0.000	0	0	0.000	0.000	/var/lib/ceph/osd/ceph-12

Figure 9 - PerfAccel IOPS Trace Of A Storage Node

This analytical information goes a step ahead and displays various data like :

a. Amount of I/O done – both in terms of Number of Ops as well as Gigabytes
   We can see here that there is negligible write ops, which tells us that the workload running until now is read-intensive .

b. I/O Throughput for each OSD
   Here we can see how the throughput from PerfAccel cache is much higher than the source

c. I/O Latency for each OSD
   Here also we can see that cached read latencies are much lesser than source latencies.

We can easily figure out which OSD is performing better and which one is choked or not used at all. For ex. here OSD ceph-9 has lesser read ops than other OSDs.

Brief Cache Trace (OSD Level)

This shows a more detailed summary of the I/O performed on the OSD, along with other useful metrics like the number of Ceph Objects cached, and the number of PerfAccel cache segments representing those Ceph objects. This analytical information also contains various stats like hits and misses of read, write, attribute read, attribute write and symlink operations done on the OSD.

```
------------------------------------------------
* ceph0: PERFACCEL CACHE STATISTICS *
------------------------------------------------
Files In Cache           : 10740
File Segments Cached     : 352815
File Segments Uncached   : 331341
File Segments Present    : 21474
Read Hits                : 12409175
Read Misses              : 1
Write Misses             : 0
Write Hits               : 4
Attribute Read Hits      : 24238
Attribute Read Misses    : 2
Attribute Write Hits     : 4
Attribute Write Misses   : 0
Links Cached             : 1
Link Hits                : 0
Link Misses              : 0
Link Cache Usage         : 4.0 KB
Total Cache Usage        : 42.4 GB (44.68%)
Source Cache Capacity    : 95.0 GB
Pending I/O requests     : 0
Misses During Cacheify   : 0
Max Cache Usage          : 42.4 GB
Total Cacheify           : 4.0 KB
File Cache Limit         : 100%
Cache Uptime             : 205 hrs 54 mins 58 secs
```

Figure 10 - PerfAccel Brief Trace of a Ceph OSD

Analytics through PerfAccel Ceph Analyzer

Now let's see the analytical information provided by the PerfAccel Ceph analyzer

Device Level View

This analytical information gives a top view of all of the VM disk images that are in use.

We can easily see here the percentage of the disk that was accessed, cached, and the measure of read hits or write hits that has happened on the devices. This gives a fair idea of what's going on with the disks in the Ceph cluster as a whole.

PerfAccel Cache Distribution (Device View) for Ceph Rados Block Devices

Ceph Rados Block Device	Size	Cached	R Hits	W Hits
0. volume-00ff159d-a95d-4d7e-877e-7dd69b7ea5c7	[60.00 GB]	20.61 GB [34.36 %]	16738741	0
1. volume-06397dc3-081e-49e1-977e-ad08697861ef	[60.00 GB]	20.67 GB [34.45 %]	16576525	0
2. volume-098b700f-8302-4aac-862d-245d0d8ccd4d	[60.00 GB]	20.70 GB [34.50 %]	16641491	0
3. volume-0d46202e-e1fb-45aa-84bd-19d3a0115e15	[60.00 GB]	20.66 GB [34.43 %]	16600641	0
4. volume-117b511e-63e6-44bc-908f-68e60dcc7b24	[60.00 GB]	21.17 GB [35.29 %]	16657638	0
5. volume-11b26b40-fe7a-4b73-afad-ccde7840d366	[60.00 GB]	20.65 GB [34.42 %]	16889202	0
6. volume-1920eccc-c11d-42ee-b05c-516d4cf657c1	[60.00 GB]	20.75 GB [34.58 %]	16714537	0
7. volume-198d6c88-dac9-4bff-b0d2-f45b58c011fb	[60.00 GB]	20.45 GB [34.08 %]	16680509	0
8. volume-199b971b-cf24-452c-b96a-878964e86c93	[60.00 GB]	20.76 GB [34.60 %]	16812233	0
9. volume-29ce2b25-2af9-4dc8-a932-a7d6d8f6f5d7	[60.00 GB]	22.05 GB [36.75 %]	16851674	0
10. volume-2f7870f3-9b63-4945-984f-94d0bda03cad	[60.00 GB]	22.43 GB [37.39 %]	16928602	0

Figure 11 - PerfAccel Ceph Analyzer Device View For All VM Disks

Node Level View

This is another data point that gives us a top view of all of the Ceph storage nodes that are serving the VM disk images. It also tells us about the percentage of Ceph objects cached, amount of cache used in the node and number of read and write hits from the cache. This gives a fair idea of what's going on with each node of the Ceph cluster.

We can now answer things like which node is heavily accessed, which node is hardly accessed, which is the most read intensive node and other similar questions.

PerfAccel Cache Distribution (Node View) for Ceph Rados Block Devices

Ceph Node	Ceph Objects	Cached Size	R Hits	W Hits
10.241.123.93	50575 [23.32 %]	197.56 GB [23.32 %]	156345299 [23.35 %]	0 [0.00 %]
10.241.123.94	59334 [27.36 %]	231.77 GB [27.36 %]	183050317 [27.34 %]	0 [0.00 %]
10.241.123.95	48510 [22.37 %]	189.49 GB [22.37 %]	149676230 [22.36 %]	0 [0.00 %]
10.241.123.96	58414 [26.94 %]	228.18 GB [26.94 %]	180441944 [26.95 %]	0 [0.00 %]

Figure 12 - PerfAccel Ceph Analyzer Node View For Storage Nodes

VM Disk Level - Brief Summary

After the device level and node level analytics, we can further drill down to individual disks. The first one in this is the brief summary analytical info about any individual VM disk.

This gives us a good amount of information about the disk like:

a. VM name to which it's attached

b. Percentage of disk that is cached

c. Percentage of cached Ceph objects for this disk

d. Cached read and write hits that has happened on the disk

e. Number of nodes where this disk is being served from (or nodes which contain the objects for this disk)

f. Percentage distribution of cached Ceph objects, read and write hits for this disk across nodes.

With these metrics we can easily answer things like :

a. Is the disk equally distributed across nodes?

b. How the I/O on the disk is distributed across nodes

c. Nature of I/O that is happening on the disk i.e. read or write

d. Access pattern of the I/O, like whether disk is fully accessed or accessed in certain areas only

```
PerfAccel Cache Distribution (Brief View) for Ceph Rados Block Devices
----------------------------------------------------------------------

1. volume-00ff159d-a95d-4d7e-877e-7dd69b7ea5c7
=============================================================================================
Ceph Storage Pool   : rbd
Attached to Host:Vm : 10.241.123.104:instance-00000017
RBD Image Size      : [ 60.00 GB]   Object Size : [4.00 MB]   Total Objects : [15360]

PerfAccel Stats
Cached Size         : 20.61  GB [ 34.36 %]
Cached Ceph Objects : 5277      [ 34.36 %]
Total Read Hits     : 16738741
Total Write Hits    : 0
Cached in Nodes     : 4 Nodes

-----------------------------------------------------------------------------------------------------
Node                    Ceph Objects      Cached Size          R Hits               W Hits
-----------------------------------------------------------------------------------------------------
10.241.123.93         1223 [  7.96 %]   4.78 GB [  7.96 %]     3891721 [ 23.25 %]          0 [  0.00 %]
10.241.123.94         1440 [  9.38 %]   5.63 GB [  9.38 %]     4539009 [ 27.12 %]          0 [  0.00 %]
10.241.123.95         1161 [  7.56 %]   4.54 GB [  7.56 %]     3690559 [ 22.05 %]          0 [  0.00 %]
10.241.123.96         1453 [  9.46 %]   5.68 GB [  9.46 %]     4617452 [ 27.59 %]          0 [  0.00 %]
```

Figure 13 - PerfAccel Ceph Analyzer Brief Summary For VM Disk

VM Disk Level - Offset Distribution

This analytical data has some information common to the brief summary info. But the one that makes it stand out is the offset distribution. Here the whole disk is divided into a given number of zones (10 in this example) and the I/O stats are displayed for each zone.

For example, in our case the disk is 60GB and we have specified 10 zones in the analyze command. So, each zone will cover 6GB and we can see all of the information distributed over each 6GB area of the disk. And if we had given 20 zones, then we would have had 3 GB zones.

This is very crucial when we need to know the behavior of the I/O happening on the disk or in other words, the I/O pattern of the Application, inside the VM, which is using this disk.

For each zone, it displays the number, percentage and size of Ceph objects cached, number and percentage of read and write hits that has happened.

In Figure 13 we can see that:

a. I/O has not crossed beyond 24GB of the disk.

b. The first 3 zones are fully cached, but the 4th zone i.e. disk area from 18GB to 24GB is partially cached.

c. The read hits are high on the first zone as compared to other zones, so the 1st 6GB of the disk is accessed more frequently than other areas.

```
PerfAccel Cache Distribution (Offset Dist. View) for Ceph Rados Block Devices
-------------------------------------------------------------------------------

1. volume-00ff159d-a95d-4d7e-877e-7dd69b7ea5c7
================================================================================================
Ceph Storage Pool   : rbd
Attached to Host:Vm : 10.241.123.104:instance-00000017
RBD Image Size      : [ 60.00 GB]   Object Size : [4.00 MB]   Total Objects : [15360]

PerfAccel Stats
Cached Size         : 20.61  GB [ 34.36 %]
Cached Ceph Objects : 5277      [ 34.36 %]
Total Read Hits     : 16738741
Total Write Hits    : 0
Cached in Nodes     : 4 Nodes

PerfAccel Cached Offset distribution
-----------------------------------------------------------------------------------------------------------
[ Offset Range ]              Ceph Objects      CachedSize                 Rhits                    Whits
-----------------------------------------------------------------------------------------------------------
[   0.00  B -   6.00 GB ] :   1536 [ 29.11 %]    6.00 GB [ 29.11 %]    5726262 [ 34.21 %]          0 [  0.00 %]
[   6.00 GB -  12.00 GB ] :   1536 [ 29.11 %]    6.00 GB [ 29.11 %]    4719529 [ 28.20 %]          0 [  0.00 %]
[  12.00 GB -  18.00 GB ] :   1536 [ 29.11 %]    6.00 GB [ 29.11 %]    4719986 [ 28.20 %]          0 [  0.00 %]
[  18.00 GB -  24.00 GB ] :    669 [ 12.68 %]    2.61 GB [ 12.68 %]    1572964 [  9.40 %]          0 [  0.00 %]
[  24.00 GB -  30.00 GB ] :      0 [  0.00 %]    0.00  B [  0.00 %]          0 [  0.00 %]          0 [  0.00 %]
[  30.00 GB -  36.00 GB ] :      0 [  0.00 %]    0.00  B [  0.00 %]          0 [  0.00 %]          0 [  0.00 %]
[  36.00 GB -  42.00 GB ] :      0 [  0.00 %]    0.00  B [  0.00 %]          0 [  0.00 %]          0 [  0.00 %]
[  42.00 GB -  48.00 GB ] :      0 [  0.00 %]    0.00  B [  0.00 %]          0 [  0.00 %]          0 [  0.00 %]
[  48.00 GB -  54.00 GB ] :      0 [  0.00 %]    0.00  B [  0.00 %]          0 [  0.00 %]          0 [  0.00 %]
[  54.00 GB -  60.00 GB ] :      0 [  0.00 %]    0.00  B [  0.00 %]          0 [  0.00 %]          0 [  0.00 %]
```

Figure 14 - PerfAccel Ceph Analyzer Offset Distribution for VM Disk

VM Disk Level - VM Disk Detailed View

a. This analytical data gives us detailed information about the disk and I/O done on it.

b. In the beginning, it displays the basic information related to the disk, as shown in the brief summary. But, after that the real metrics are seen.

c. The analytical data is showing the I/O stats at the lowest level i.e. for each and every cached Ceph object of the disk. It is displaying various metrics like Read Hits, Write Hits, Last Access Time & Resident time in cache, for any Ceph object.

Now, as the objects belonging to this disk are distributed across Nodes, so it starts with displaying a brief summary of this disk in a particular node and then continues displaying the detailed information of each object present in that node. It does this for all the Nodes where the objects are present.

As each Ceph object represents a certain area of the disk, so it displays all the metrics against the real offset of the disk to which the object corresponds to.

With this kind of analytical information, we can find out the I/O pattern & signature at the offset level.

```
PerfAccel Cache Distribution (Detail View) for Ceph Rados Block Devices
-------------------------------------------------------------------------

1. volume-00ff159d-a95d-4d7e-877e-7dd69b7ea5c7
=============================================================================================
Ceph Storage Pool    : rbd
Attached to Host:Vm  : 10.241.123.104:instance-00000017
RBD Image Size       : [ 60.00 GB]    Object Size : [4.00 MB]    Total Objects : [15360]

PerfAccel Stats
Cached Size          : 20.61  GB [ 34.36 %]
Cached Ceph Objects  : 5277       [ 34.36 %]
Total Read Hits      : 16738741
Total Write Hits     : 0
Cached in Nodes      : 4 Nodes

--------CEPH NODE : 10.241.123.93---------

No. of Ceph Objects : 1223         [  7.96 %]
Image Size Cached   :  4.78 GB     [  7.96 %]
Read Hits           : 3891721      [ 23.25 %]
Write Hits          : 0            [  0.00 %]

-------------------------------------------------------------------------
         Offset        Size       RHits       WHits    LaT(sec)     RT(sec)
-------------------------------------------------------------------------
     1207959552  4198400        4115           0        1058        7215
     2772434944  4194304        4114           0        1047        7032
     1920991232  4198400        4114           0        1053        7119
     1765801984  4194304        4114           0        1054        7141
     2512388096  4194304        4113           0        1048        7053
     1233125376  4198400        4113           0        1058        7208
      377487360  4194304        4112           0        1066        7261
     1673527296  4198400        4112           0        1055        7161
     3292528640  4198400        4112           0        1044        6982
     2973761536  4194304        4112           0        1045        7014
```

Figure 15 - PerfAccel Ceph Analyzer Detailed View For VM Disk

"PerfAccel can be rapidly deployed at scale in OpenStack environments, to achieve greater I/O visibility and Performance of Virtual Machines and Applications"

Conclusion

In this reference architecture, we were able to successfully validate the performance enhancements and the rich analytics provided by PerfAccel in a Ceph cluster. The results and analytical data obtained from the 4 node Ceph cluster and 2 node OpenStack setup on Intel Xeon and Atom processor-based servers and using Intel SSDs, clearly demonstrated that PerfAccel is a promising solution for Ceph environments where there is a need for performance as well as a fine grained and rich set of analytical data.

Call To Action

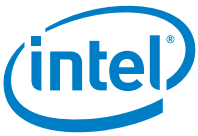
Contact us at info@datagres.com to schedule a demo or visit us at www.datagres.com/download, to start a free trial.

About Datagres:

Datagres provides software that helps companies visualize, control and accelerate their application performance using deep storage intelligence. Datagres' flagship product PerfAccel is a very powerful analytics driven software solution that operates at a file level and can show the exact IO pattern of an application data access especially in a scale-out grid environment. As a result, it can provide an effective way of controlling IOs and also accelerate for higher throughput and lower latencies using high-performance SSD devices.

The company is headquartered in Palo Alto, California and is venture-backed by Nexus Venture Partners

For more information, visit www.datagres.com



Disclaimers

Printed in USA Please Recycle