Ampere® Altra® Family 64-Bit Multi-Core Processors

Ampere® Arm® Processors for Ceph – White Paper

February 22, 2023

Document Issue 1.00

Contents

1. Executive Summary 6
1.1 Scope and Audience 6
2. Ceph Architecture Overview 7
2.1 Ceph Components 7
2.2 Design Considerations for Ceph 8
3. Ampere Tests and Validation of Ceph 9
3.1 Ampere Test Bed Architecture and BOM 9
3.2 Hardware Components 9
3.3 Software Components 10
3.4 Supermicro Servers 10
3.5 KIOXIA CD8 NVMes 10
3.6 Pre-Installation Checks and Best Practices 10
3.6.1 BIOS Changes 10
3.6.2 System Memory 10
3.6.3 PCIe Bifurcation 10
3.6.4 Network Interface Cards (NICs) 10
3.6.5 Sysctl 11
3.6.6 MTU Settings for Cluster Network 11
3.6.7 Bandwidth Check Between Servers and Clients 11
3.6.8 OSD Disks 11
3.6.9 Run Baseline Tests on Disks 11
3.6.10 FIO (Flexible I/O) for Arm 11
4. Ceph Installation 12
5. Performance Testing 13
5.1 Block Storage Tests 13
5.1.1 Single FIO Process on Cluster 13
5.1.2 Multiple FIO Processes on Cluster 14
5.2 Object Storage Tests 14
5.2.1 Summary 15
6. Continuous OSD Stress Tests 16
6.1 Summary 16
7. Interoperability tests – Heterogeneous CPU Topology 17
8. Power Consumption 18
8.1 Power Consumption: Four-node Cluster with Ampere Processors 18
8.1.1 Summary 18

Contents (continued)

8.2 Power Consumption: Ampere vs AMD Nodes 18
8.2.1 Summary 18
9. Migrating an x86 Cluster to Ampere Nodes 19
9.1 Node and Components Swap-Out 19
9.1.1 Steps to be Followed 19
9.2 Node Swap-Out and Disk Reuse 20
9.2.1 Steps to be Followed 20
9.2.2 Observations 21
10. Ceph Tracker Issues 22
11. Conclusion 23
12. Appendix 24
12.1 BIOS 24
12.2 Linux 24
12.3 /etc/sysctl.conf 24
12.4 /etc/security/limits.conf 24
12.5 Ceph 25
12.6 FIO Commands 25
12.7 COSbench – Workload-config.xml 26
Document Revision History 27

Figures

Figure 1: Ceph Architecture 7
Figure 2: Ampere Test Bed Architecture 9
Figure 3: Cluster Configuration for Block Storage Tests 13
Figure 4: Test Results with Single FIO Processes 13
Figure 5: Test Results with Multiple FIO Processes 14
Figure 6: Dashboard Snapshot with Multiple FIO Processes 14
Figure 7: Sample Results from COSBench and Ceph Dashboard 15
Figure 8: Test Results for GETS and PUTS on the COSBench Tool 15
Figure 9: Results for Continuous OSD Stress Tests 16
Figure 10: Heterogeneous Ceph Cluster with x86 and aarch64 Nodes 17
Figure 11: Power Consumption Graphs using IPMI Measurements 18
Figure 12: Power Consumption Graphs for Ampere and AMD Nodes 18
Figure 13: Swapping Out an AMD x86 Node with an aarch64 Ampere Node 19
Figure 14: Dashboard Snapshot for Node Swap-Out and Disk Reuse 20
Figure 15: Read and Write IOPS During Node Swap-Out and Disk Reuse 21

Tables

Table 1: Ampere Server Inventory 9

Table 2: x86 Server Inventory... 9

1. Executive Summary

Ceph is a reliable, scalable, and fault-tolerant storage solution that can uniquely deliver block, file, and object storage in one unified platform. It is open-source and cost-effective for modern enterprises and cloud environments.

Ampere's Altra® and Altra Max® processors are designed to deliver high performance with single-threaded cores that run at consistently high frequencies and at large low-latency private caches. The architecture allows for high utilization and consistent performance even under heavy loads. These processors families were built from the ground up with an innovative scale-out architecture, featuring high core counts and efficient scaling. They are also highly power-efficient compared to x86 processors.

This document offers a comprehensive examination of a Ceph cluster using Ampere Arm processors, highlighting their effectiveness in block and object storage. It includes performance data collected from a four-node test bed, examines compatibility with x86 processors, highlights power consumption, and provides methods for migrating a Ceph cluster from x86 to Arm processors.

1.1 Scope and Audience

In this writing, we cover the process of establishing a four-node cluster using Ampere processors, including the implementation of various tips, techniques, and tuning methods. Performance data from the test bed is included, as well as guidance for migrating an existing x86 Ceph cluster to Ampere Arm processors. Additionally, this document provides general guidelines and optimization suggestions for Ceph clusters. It should be noted, however, that these values and parameters should not be considered as definitive or optimized.

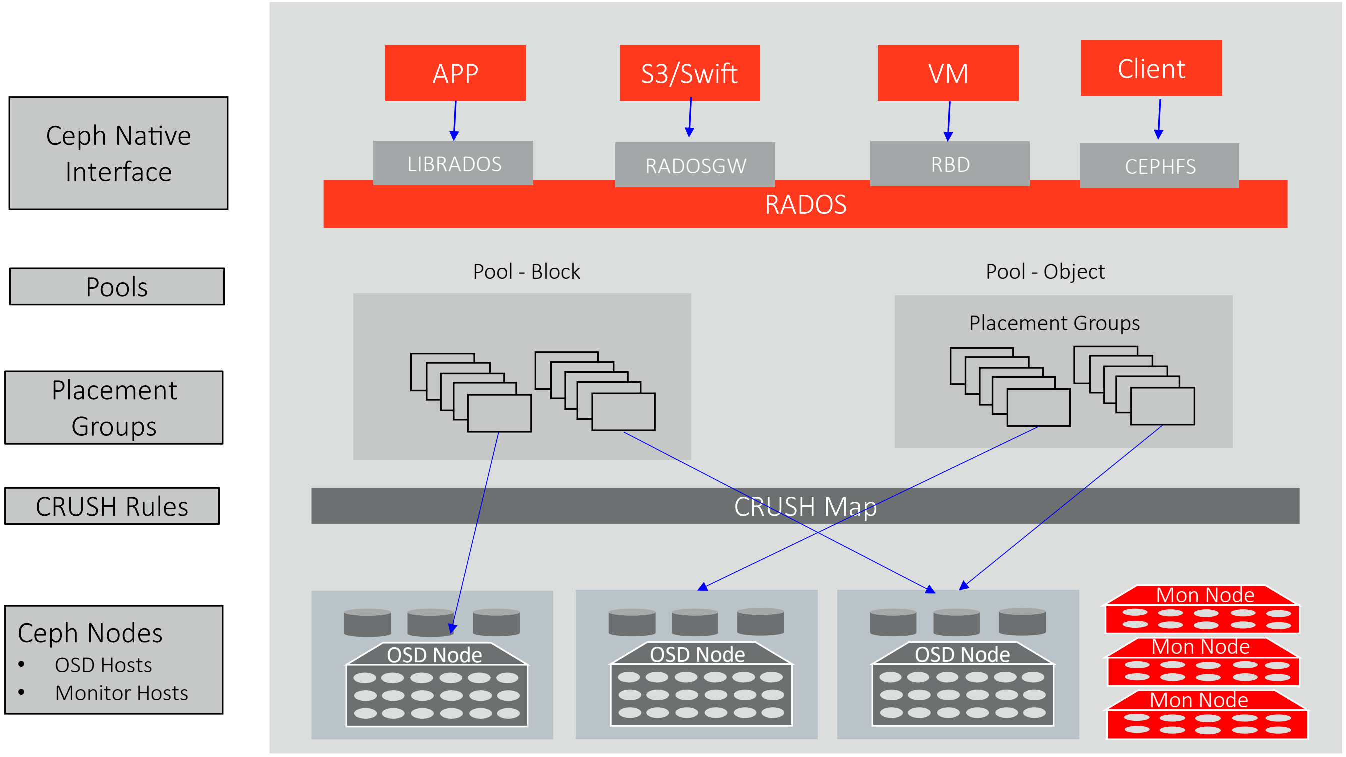
The content of this document is intended to enable Sales Engineers, Cloud Storage providers, IT and Cloud Architects, and end-user customers to take advantage of Ampere Arm servers.

2. Ceph Architecture Overview

Ceph is an open-source storage platform that offers scalability, fault tolerance, and high performance. It supports petabytes of storage and uses its own CRUSH algorithm to separate data and metadata operations, eliminating the need for file allocation tables. Ceph also utilizes devices for not only data access, but also for serialization, replication, and failure detection, making it a powerful alternative to traditional storage solutions.

Ceph is a popular choice among cloud environments, OpenStack, Kubernetes, and other container-based platforms. It offers different interfaces to cater to various storage needs within a single cluster, eliminating the need for multiple storage solutions or specialized hardware, thereby reducing management overhead.

Figure 1: Ceph Architecture



2.1 Ceph Components

Ceph is a distributed storage system that comprises several components that work together to provide a scalable, fault-tolerant, and high-performance storage solution. The main components of a Ceph cluster include:

- Ceph Monitors (ceph-mon): These maintain the cluster map, track the health of the nodes, manage the cluster configuration, and are responsible for authenticating clients. It is recommended to have an odd number of monitors in the Ceph cluster for production deployments.
- Ceph Manager (ceph-mgr): These handle runtime metrics, provide a dashboard, and offer an interface to external monitoring systems.
- Ceph Object Storage Devices (ceph-osd): These store the data in the cluster and replicate it to different disks, nodes, or racks based on the configuration. They also rebalance and recover data as needed. A reasonable amount of CPU capacity is needed for OSD daemons for block storage doing small block IOs. If you plan to separate Monitor and OSD nodes, it is recommended to allocate more cores to OSD nodes. On the other hand, Monitors are not CPU intensive.
- RADOS Gateways (ceph-rgw): RADOS Gateways provide Swift and S3 APIs for accessing the data stored in the cluster.

- Pools: These are the logical partitions utilized in Ceph to store data objects. Pools can be created for several types of data repositories, such as block devices or object gateways, or for different user groups. In a replicated storage pool, Ceph makes multiple copies of the objects based on the replication factor. In an erasure-coded pool, the objects are divided into chunks using an n=k+m equation.
- Ceph Placement Groups (PGs): These maintain static mapping between objects and physical disks. Objects are placed into PGs, and the PGs are further mapped to OSDs. Increasing the number of PGs will reduce the variance on a per-OSD basis, but it may increase CPU and memory usage on monitors and OSD servers.
- CRUSH (Controlled Replication Under Scalable Hashing): This algorithm determines how to store and retrieve data. CRUSH maps are specified on a per-pool basis. It ensures that the replicas do not end on the same host, disk etc.

Ceph encompasses several features, such as:

- Self-healing and self-management
- Exabyte scalability
- S3 and Swift API support
- Built in security and data protection
- Reliability and availability
- Multi-datacenter support and disaster recovery
- Cost-effectiveness and web-based management

2.2 Design Considerations for Ceph

The following points can be considered while designing a Ceph cluster.

- What are the IOPS and bandwidth needs to meet the storage requirements, specifically for the application running on the Ceph cluster? Block storage requires high IOPS, while Object storage requires higher bandwidth.
- What is the recommended percentage of free space to be kept on a single node and how does it affect the calculation of disk capacity?
- What are the reliability needs, and how do they impact disk capacity calculations? Ceph has two options for reliability: Replication and Erasure Coding.
  1. Replication copies every object to one or more secondary OSDs based on the replication factor.
  2. Erasure Coding improves storage efficiency by distributing data and coding chunks.

  The effective space available for data is 6/8, which is 75% of raw capacity with 6+2 EC profile set, while in replicated mode with replication factor as 2, it is only 50% of the raw capacity. However, it has a performance penalty, as data must be split into chunks before writing or coalesced before reading. The default in Ceph is replicated pools. It is important to factor in organic growth and future expansion of the cluster when making raw disk capacity calculations.

- What is the preferred failure domain for cluster design, node, or rack level? Ceph defaults to configuring the failure domain at the node level. While failure of a single node can cause some performance impact due to backfilling operations, it ensures business continuity. However, if higher availability is desired, the failure domain can be set at the rack level in a data center by distributing the nodes across different racks and adjusting the CRUSH algorithm. Ceph will handle the rest, providing high availability in the event of a total failure, such as a power failure affecting all nodes in a physical rack.

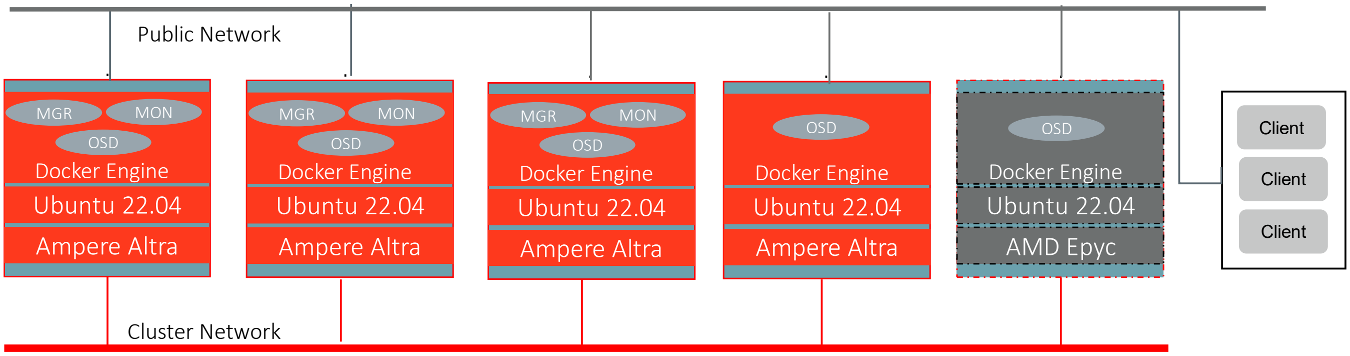
3. Ampere Tests and Validation of Ceph

Ampere Computing conducted several tests on a Ceph cluster with its Ampere Altra Arm processors and the results gathered are presented in the sections below. The tests included:

1. Block storage performance
2. Object storage performance
3. Ampere Arm and x86 interoperability in a single cluster
4. Measuring power consumption of Ampere and x86 servers in a cluster
5. Migrating an x86 cluster to Arm with no or minimal disruption

3.1 Ampere Test Bed Architecture and BOM

Figure 2: Ampere Test Bed Architecture



The test setup included four Ampere Altra processor nodes and one AMD EPYC processor node.

Block and object storage performance tests were run on the four-node Ampere cluster. On the other hand, the AMD node in conjunction with a three-node Ampere cluster was used to test interoperability and migration to and from Arm.

3.2 Hardware Components

Table 1: Ampere Server Inventory

ITEM	DESCRIPTION	QUANTITY
Model	Supermicro Mt. Hamilton ARS-110M-NR	4
CPU	Ampere Altra Q64-30, 64 cores 3.0 GHz	1
Memory	Samsung DDR4 3200 MHz 16GB DIMMS	8
Disk – Root	960 GB Samsung NVMe	1
Disks – OSD	6.4 TB KIOXIA CD8-V	3
Network Adapter	NVIDIA Mellanox CX-4	2

Table 2: x86 Server Inventory

ITEM	DESCRIPTION	QUANTITY
Model	Thinkmate Server ASUS RS500A	1
CPU	AMD EPYC 7513 32 Cores/64 Threads 2.6 GHz	1

ITEM	DESCRIPTION	QUANTITY
Memory	Samsung DDR4 3200 MHz 16GB DIMMS	8
Disk – Root	960 GB Samsung NVMe	1
Disks – OSD	6.4 TB KIOXIA CD8-V	3
Network Adapter	NVIDIA Mellanox CX-4	2

Network Switch: 1 x Arista DCS-7060SX2-48YC6-R

Additionally, 8 clients were used as load generating machines that are not listed above.

3.3 Software Components

- Operating System: Ubuntu Jammy Jellyfish 22.04
- Ceph Quincy 17.2.5 on Containers

3.4 Supermicro Servers

Supermicro Mt. Hamilton Single Socket 1U servers with Ampere Altra Q64 processor were used. For more details, refer to this link.

3.5 KIOXIA CD8 NVMes

KIOXIA Data Center NVMe SSDs CD8 (PCIe Gen4x4) Series were used in the Ceph test environment. The CD8 Series which are equipped with KIOXIA's 5th generation BiCS FLASH™ 3D flash memory technology, firmware and a controller developed by KIOXIA, are suitable for cloud-based applications run in an industry standard server environment to be scaled out in a cloud. These SSDs include data protection with power-loss protection (PLP) and encryption technology options (for SIE and SED models) to increase safety and security.

3.6 Pre-Installation Checks and Best Practices

The following sections outline recommended practices for optimizing performance in a Ceph system.

3.6.1 BIOS Changes

Although most of the BIOS changes recommended in the *Appendix* are typically set as default, it is suggested to verify them as a best practice.

3.6.2 System Memory

Ensure that all memory DIMMs have the same capacity and speed. The test bed consisted of 8 DIMMs per node, each with 16GB of memory.

3.6.3 PCIe Bifurcation

PCIe bifurcation is x4+x4+x4+x4 for the NVMe disks being used. If using KIOXIA disks, you can check the link status and speed with the following command:

```
sudo lspci -vvv | grep -i KIOXIA -A 50 | grep -i LnkSt | grep -i Speed
```

3.6.4 Network Interface Cards (NICs)

It is essential to ensure that you have the latest NVIDIA Mellanox firmware update on your cards. You can check this by running the command `mlxup –query`. The `mlxup` tool can be downloaded from NVIDIA.

3.6.5 Sysctl

Recommended changes applied to sysctl are included in the *Appendix*.

3.6.6 MTU Settings for Cluster Network

If you have a separate cluster network, it is recommended to have 9000 MTU for this network.

3.6.7 Bandwidth Check Between Servers and Clients

Ensure that the rated network bandwidth between servers and clients is achieved using iperf.

3.6.8 OSD Disks

While it is feasible to use disks of different sizes in Ceph, the weight assigned by the CRUSH algorithm will be proportional to the size of the disks. The performance of the disks also depends on the make and model. To achieve optimal performance, it is recommended to avoid mixing disks of different makes and sizes for OSDs in a pool. Additionally, it is important to precondition the disks before using them in Ceph. For instance, to ensure optimal performance for a workload with 4k block size, precondition the disks using the same block size.

3.6.9 Run Baseline Tests on Disks

It is recommended to run baseline tests on the disks to ensure that they are close to the manufacturer's specification.

3.6.10 FIO (Flexible I/O) for Arm

As of this document's writing, the version of FIO installed on the operating system did not support RBD on aarch64 architecture. To test against RBD devices, the FIO source code was downloaded from GitHub and compiled.

4. Ceph Installation

Cephadm was used to install Ceph (version Quincy 17.2.5) on Ubuntu operating systems using Docker containers. The specific steps of the installation process are not discussed in this paper, but comprehensive instructions for installing Ceph can be found at this link.

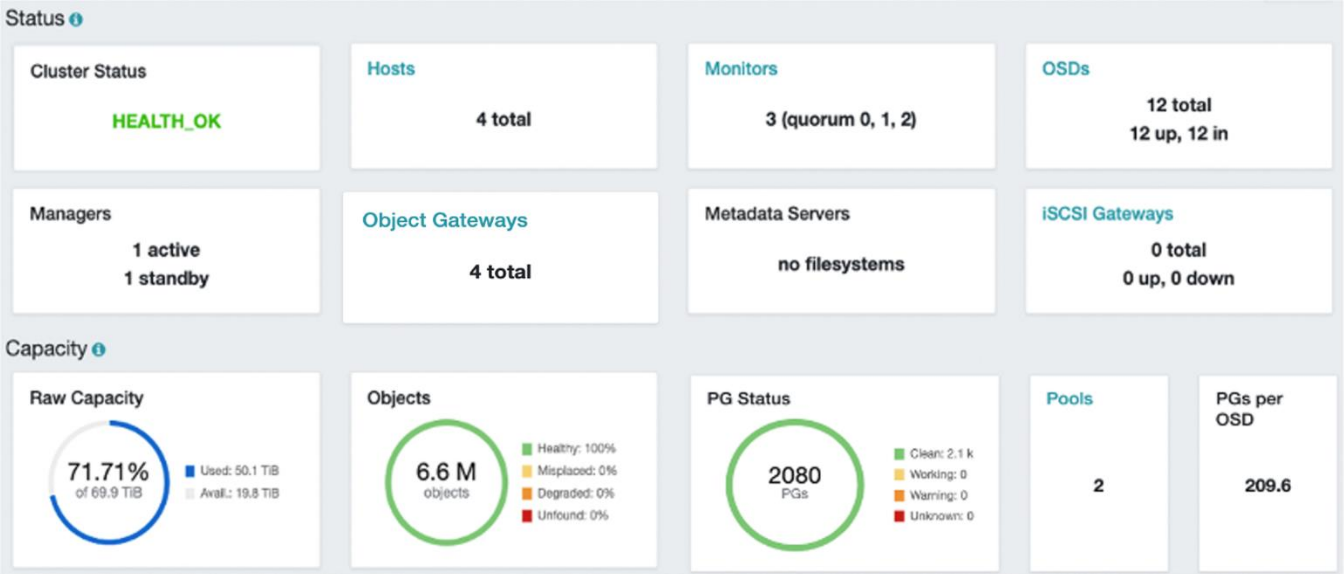
5. Performance Testing

Optimizing Ceph performance can be challenging due to the vast number of options that can be adjusted between Ceph, the Linux kernel, and RocksDB. It is impractical to test all variations of these options, but an effort was made to alter few key parameters, which are listed in the *Appendix* in this document. Our focus was on Block and Object storage in this exercise; no testing has been conducted as of yet regarding Ceph's file storage functionality.

5.1 Block Storage Tests

Block storage tests were performed on a four-node cluster, and FIO tests were run from client nodes. Before conducting the tests, the cluster was configured as shown in *Figure 3*.

Figure 3: Cluster Configuration for Block Storage Tests



Status

Cluster Status
HEALTH_OK

Hosts
4 total

Monitors
3 (quorum 0, 1, 2)

OSDs
12 total
12 up, 12 in

Managers
1 active
1 standby

Object Gateways
4 total

Metadata Servers
no filesystems

iSCSI Gateways
0 total
0 up, 0 down

Capacity

Raw Capacity
71.71%
of 69.9 TiB
Used: 50.1 TiB
Avail.: 19.8 TiB

Objects
6.6 M
objects
Healthy: 100%
Misplaced: 0%
Degraded: 0%
Unfound: 0%

PG Status
2080
PGs
Clean: 2.1 k
Working: 0
Warning: 0
Unknown: 0

Pools
2

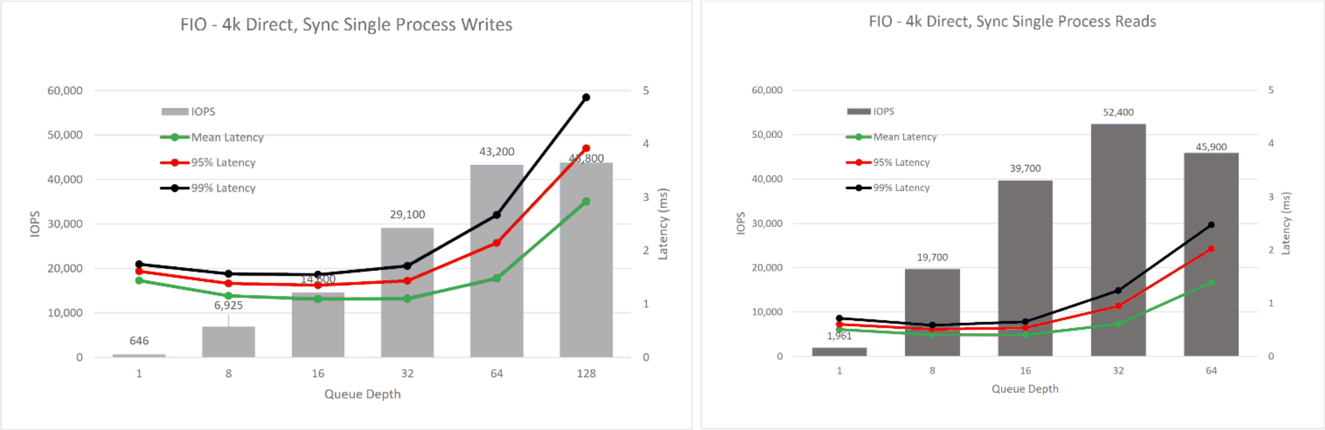
PGs per OSD
209.6

Each node had 3 OSDs. Approximately 6.6 million 4k objects were created with a replication factor of 2, utilizing 70% of the raw disk space.

5.1.1 Single FIO Process on Cluster

Results with a single FIO process are shown in *Figure 4*.

Figure 4: Test Results with Single FIO Processes



FIO Commands are included in the *Appendix*.

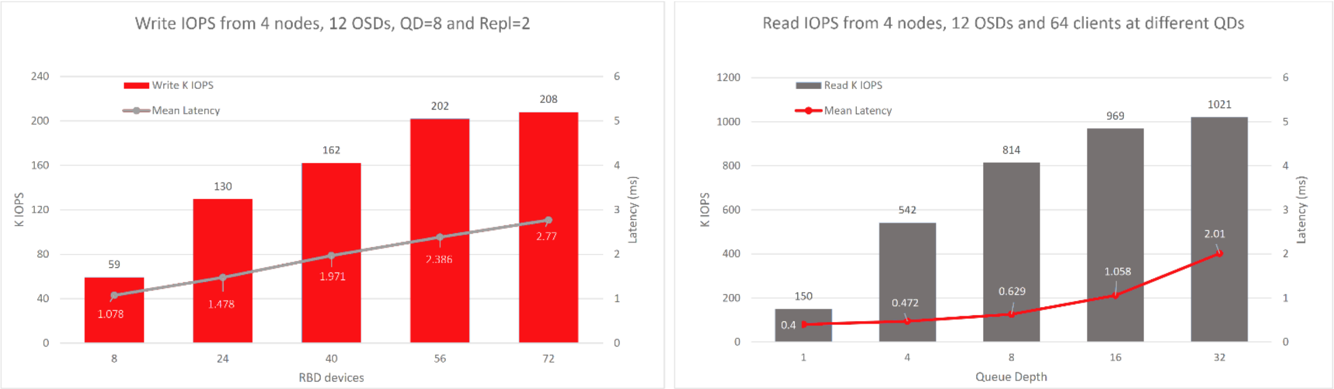
5.1.1.1 Summary

- The optimal queue depths for read and write operations with a single client were determined by conducting tests with adjusted queue depths. These results were applied when multiple clients were added.
- We achieved approximately 43k write IOPS with a mean latency of 1.4ms.
- We achieved approximately 52k read IOPS with a mean latency of 0.6ms.
- Further increasing the queue depth resulted in diminishing returns for IOPS and an increase in tail latencies.

5.1.2 Multiple FIO Processes on Cluster

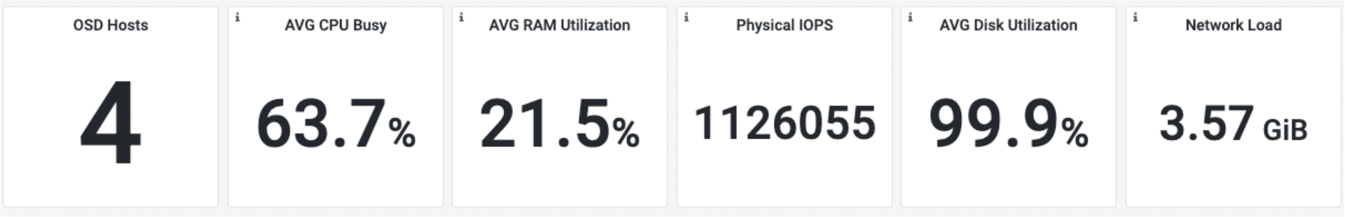
Tests were repeated using multiple clients and the results for writes and reads were as shown in *Figure 5*.

Figure 5: Test Results with Multiple FIO Processes



The dashboard snapshot, with 72 processes and 12 OSDs, looked as shown in *Figure 6*.

Figure 6: Dashboard Snapshot with Multiple FIO Processes



OSD Hosts	AVG CPU Busy	AVG RAM Utilization	Physical IOPS	AVG Disk Utilization	Network Load
4	63.7%	21.5%	1126055	99.9%	3.57 GiB

5.1.2.1 Summary

- The RBD (RADOS Block device) devices were pre-populated with data.
- Write IOPS tests were run at QD=8 while varying the number of clients.
- Using the optimal number of clients obtained from writes, read tests were performed by varying the queue depth.
- Six client machines were used in the test bed, each running FIO against each RBD device.
- We achieved approximately 200k write IOPS with a mean latency of 2.3ms from 12 OSDs in the cluster.
- We also achieved around 970k read IOPS with a mean latency of 1ms from the same cluster at a queue depth of 16.
- Increasing the number of RBDs and/or clients resulted in only marginal improvements in IOPS.

5.2 Object Storage Tests

After initial testing with RADOS bench, the object storage performance was evaluated using the COSBench tool. The setup was configured as follows:

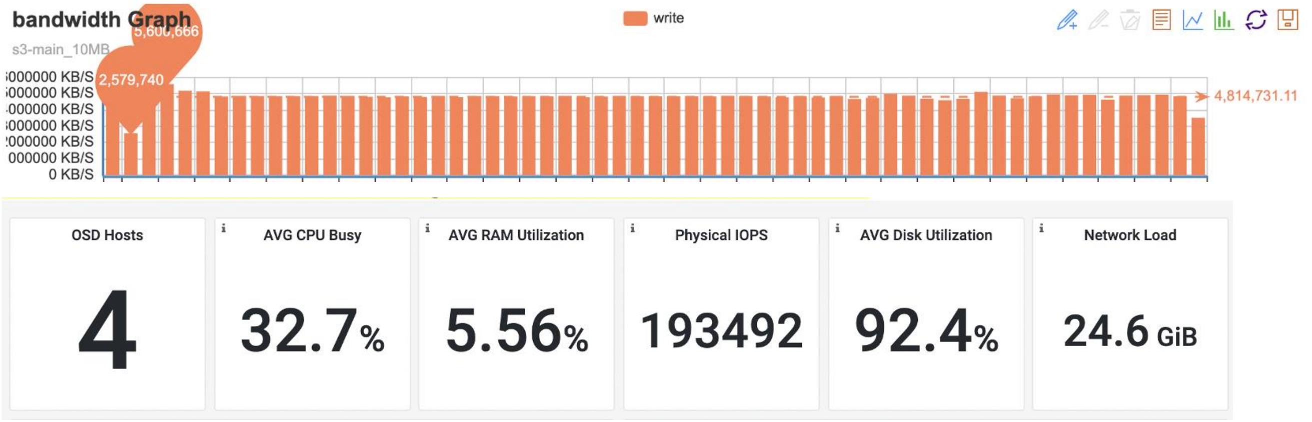
- Added object store with Ceph orchestrator and configured two gateways per node.

```
ceph orch apply rgw objstore --port=8000 '--placement=label:objstore count-per-host:2'
```

- Created a HAproxy server to load balance against the RADOS Gateways.
- Configured COSBench. A sample xml file from COSBench is included in the *Appendix*.
- A single container was used for writes, while multiple of them were used for reads.

Here is a snippet from COSBench and Ceph dashboards while writing 10MB PUTS (*Figure 7*).

Figure 7: Sample Results from COSBench and Ceph Dashboard



OSD Hosts
4

AVG CPU Busy
32.7%

AVG RAM Utilization
5.56%

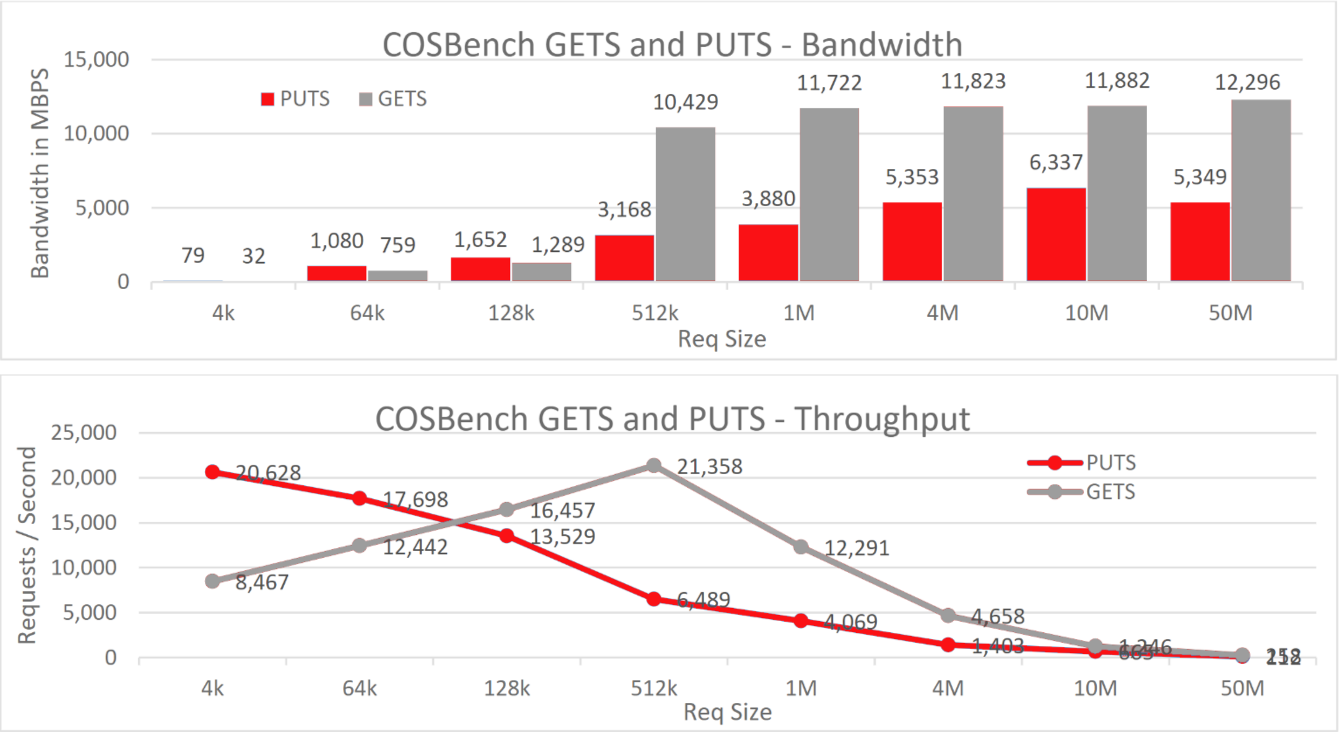
Physical IOPS
193492

AVG Disk Utilization
92.4%

Network Load
24.6 GiB

Tests using the COSBench tool were performed with block sizes ranging from 4k to 50M for both GETS and PUTS. The results, which included data on throughput and bandwidth, were plotted in the graph shown in *Figure 8*.

Figure 8: Test Results for GETS and PUTS on the COSBench Tool



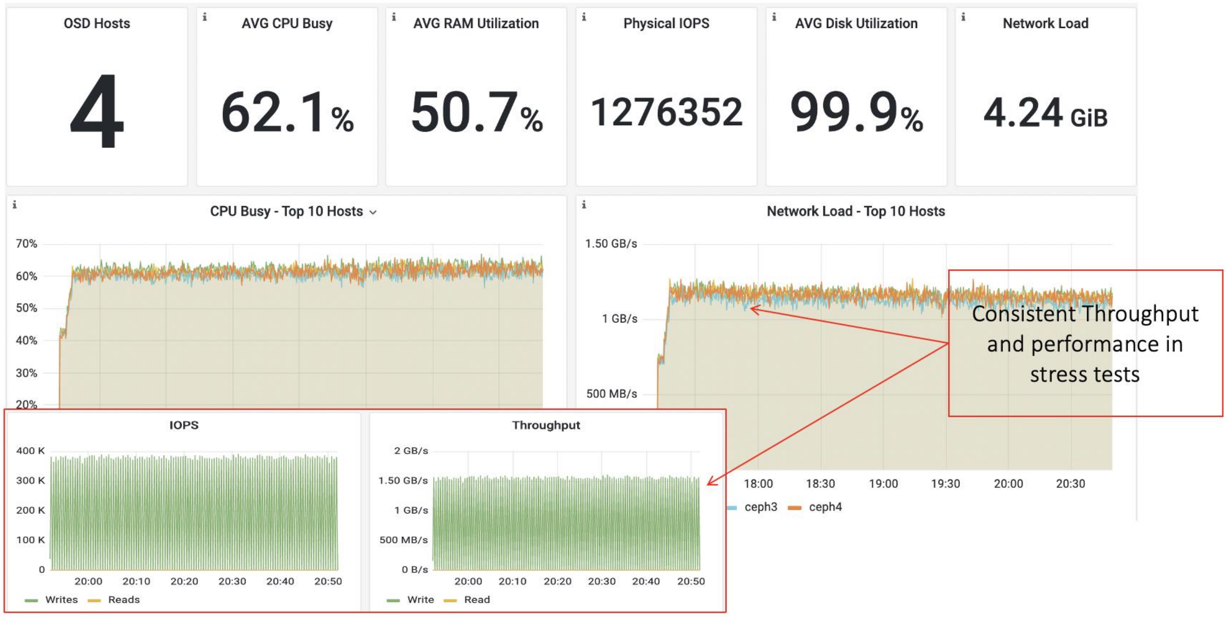
5.2.1 Summary

- The object tests were constrained by network bandwidth. The system had a network capacity of 100 Gb/s across all four nodes, resulting in a maximum bandwidth of 12,500 MB/s. As seen from the bandwidth graph, the bandwidth jumped to 12,300 MB/s at block size of 50M.
- The GETS throughput achieved a maximum of around 21k Requests/s at a block size of 512k.

6. Continuous OSD Stress Tests

Continuous stress tests were conducted on the system to assess any decline in cluster performance. This was done by monitoring both the client-side output and the Ceph dashboard for any indications while running FIO tests for six hours. This was attempted on the same four-node cluster that was used in block-storage performance tests. The recorded output is as shown in *Figure 9*.

Figure 9: Results for Continuous OSD Stress Tests



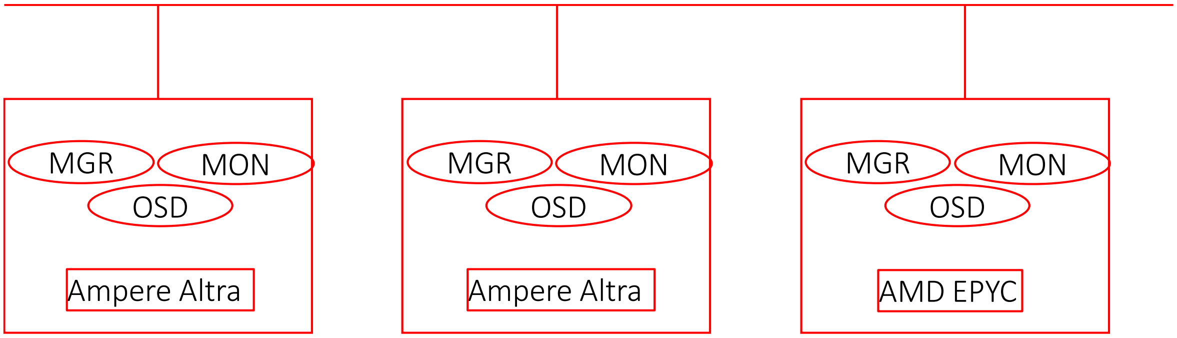
Consistent Throughput and performance in stress tests

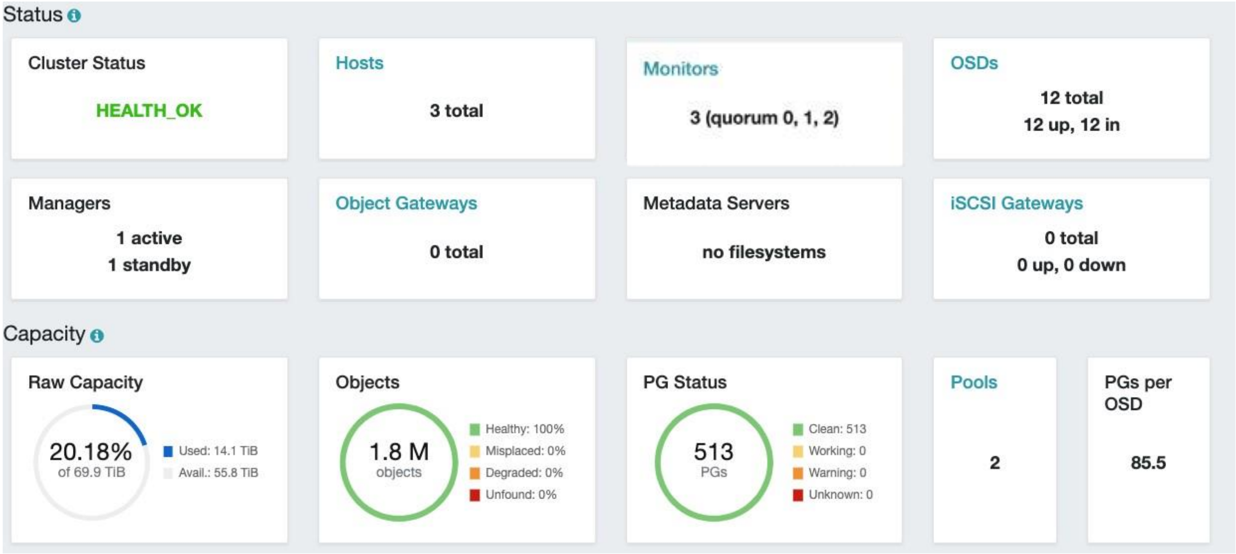
6.1 Summary

- The block storage FIO tests were run for six hours.
- No unusual activity was found in either Linux system message files or Ceph monitor and OSD logs.
- The system's CPU usage was consistently at 60-65%.
- The system's IOPS, throughput, and latencies remained stable.

7. Interoperability tests – Heterogeneous CPU Topology

Figure 10: Heterogeneous Ceph Cluster with x86 and aarch64 Nodes





Status

Cluster Status
HEALTH_OK

Hosts
3 total

Monitors
3 (quorum 0, 1, 2)

OSDs
12 total
12 up, 12 in

Managers
1 active
1 standby

Object Gateways
0 total

Metadata Servers
no filesystems

iSCSI Gateways
0 total
0 up, 0 down

Capacity

Raw Capacity
20.18%
of 69.9 TiB
Used: 14.1 TiB
Avail.: 55.8 TiB

Objects
1.8 M
objects
Healthy: 100%
Misplaced: 0%
Degraded: 0%
Unfound: 0%

PG Status
513
PGs
Clean: 513
Working: 0
Warning: 0
Unknown: 0

Pools
2

PGs per OSD
85.5

We conducted a study of a Ceph cluster composed of both x86 and aarch64 nodes. We removed two Ampere nodes and added an AMD node, resulting in a three-node cluster with monitors on all nodes. The Ampere nodes had one 64 single-core CPU each while the AMD node featured one 32 core (64 thread) CPU instead. Each node had 4 OSDs, with pre-populated data and no difference in CRUSH map between x86 and Arm nodes observed either.

FIO tests were run, and the system remained stable with no errors reported.

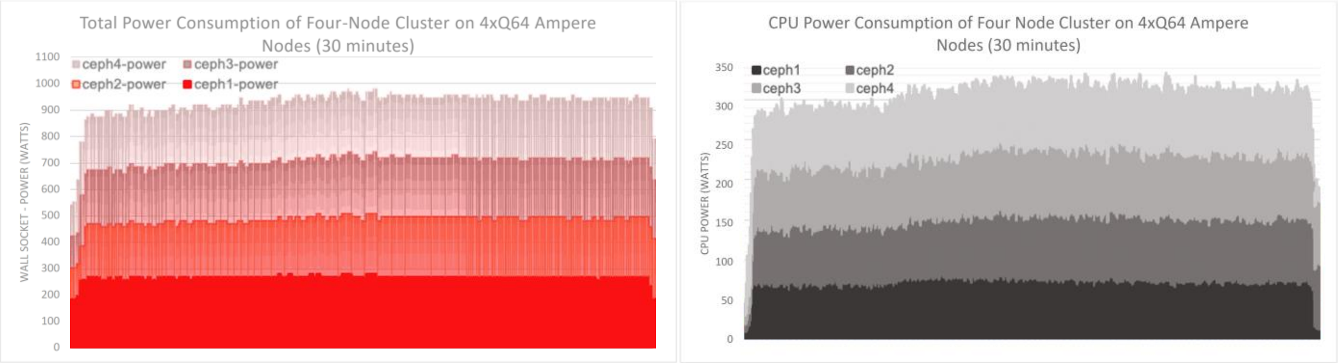
8. Power Consumption

We measured the power drawn by the Supermicro servers with Ampere Arm processors by conducting FIO tests with 4k writes on a four-node Ceph cluster. The system was stressed to deliver around 200k IOPS as noted in the performance section. We then compared the power consumption of the ASUS server with AMD EPYC 7513, which replaced one of the Ampere nodes in the four-node cluster.

We first measured the system power using a power meter on the PDU. However, we later switched to using IPMI power draw commands, as we found only a 3% difference between the two methods. The results presented in *Figure 11* are from the IPMI measurements. The total incoming power supply to each PSU was measured.

8.1 Power Consumption: Four-node Cluster with Ampere Processors

Figure 11: Power Consumption Graphs using IPMI Measurements

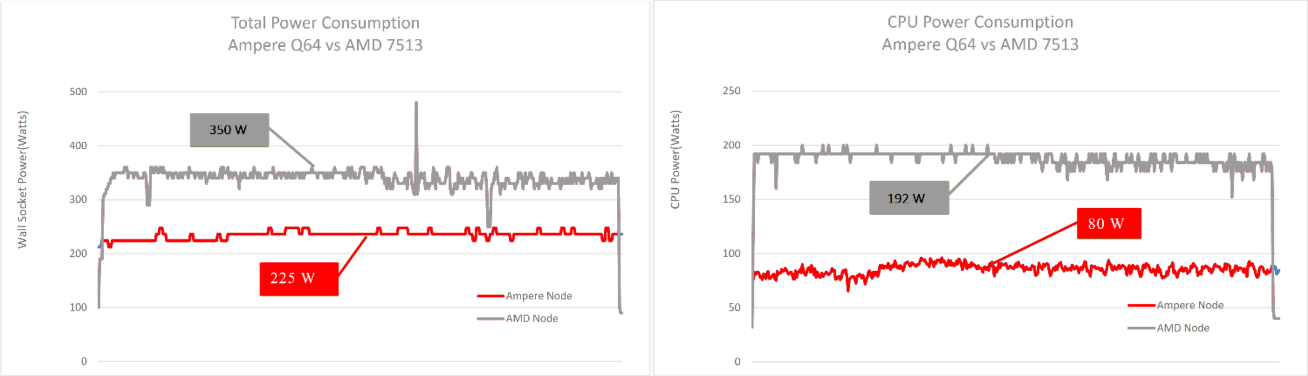


8.1.1 Summary

- We found that each Supermicro server featuring Ampere Altra processors consumed approximately 225 watts of power. The whole four-node cluster consumed roughly 900 watts.
- Both IPMI and lm-sensors on Ubuntu were used to measure the power consumption of the CPU, and both methods showed a power draw of roughly 80 watts per CPU.

8.2 Power Consumption: Ampere vs AMD Nodes

Figure 12: Power Consumption Graphs for Ampere and AMD Nodes



8.2.1 Summary

- The AMD node had a power consumption of approximately 350 watts, while the Ampere node consumed 225 watts.
- The AMD CPU had a power consumption of 192 watts, while the Ampere CPU consumed around 80 watts.

9. Migrating an x86 Cluster to Ampere Nodes

We previously tested interoperability by combining x86 and Ampere Arm nodes. We then moved on to examining how to migrate an existing x86 cluster to Ampere Arm nodes. While Ceph supports multi-site configuration and RBD mirroring, the below tests were done by replacing x86 nodes. The testing involved replacing nodes one by one with minimal disruption to the ongoing workload, such as replacing a failed node in production. Despite different CPU architectures, both systems are Linux-based and use little-endian.

9.1 Node and Components Swap-Out

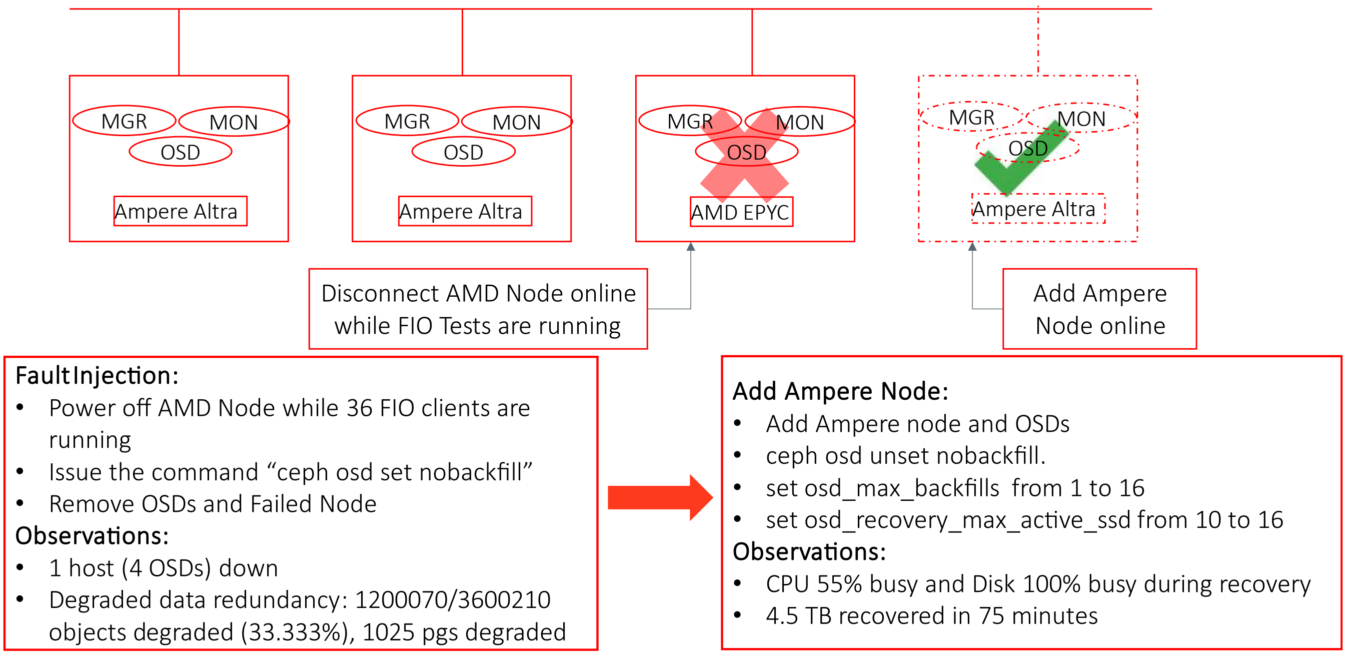
This approach involves replacing an x86 node with an Ampere Arm node. We then reinstall the operating system with aarch64 architecture and use new Ceph OSD disks.

9.1.1 Steps to be Followed

1. Set up an Ampere Arm node and install the operating system and Ceph prerequisites.
2. Disable backfilling in the cluster.
3. Power off the x86 node and use Ceph Orchestrator commands to forcefully remove the node from the cluster.
4. Connect the Ampere node (which should have the operating system already installed and the disks in place) to the cluster.
5. Format the drives as needed.
6. Use Ceph Orchestrator commands to add the new node to the storage cluster.
7. Reactivate backfilling feature.
8. Ceph will redistribute the data to the new node.

The above process was tested on a three-node cluster with one of them being an AMD node. The RBDs had data pre-populated, FIO load was started and then the AMD node was powered off. Each node had a raw data capacity of 4.5 TB when the tests were run. It took approximately 75 minutes for Ceph to return to a healthy state. The FIO jobs continued without interruption. The CPU and network loads on the cluster increased while backfilling was taking place. The backfilling operations can be adjusted using the `osd_max_backfills` parameter, which was set to 16 on the testbed.

Figure 13: Swapping Out an AMD x86 Node with an aarch64 Ampere Node



Fault Injection:
- Power off AMD Node while 36 FIO clients are running
- Issue the command “ceph osd set nobackfill”
- Remove OSDs and Failed Node

Observations:
- 1 host (4 OSDs) down
- Degraded data redundancy: 1200070/3600210 objects degraded (33.333%), 1025 pgs degraded

Add Ampere Node:
- Add Ampere node and OSDs
- ceph osd unset nobackfill.
- set osd_max_backfills from 1 to 16
- set osd_recovery_max_active_ssd from 10 to 16

Observations:
- CPU 55% busy and Disk 100% busy during recovery
- 4.5 TB recovered in 75 minutes

9.2 Node Swap-Out and Disk Reuse

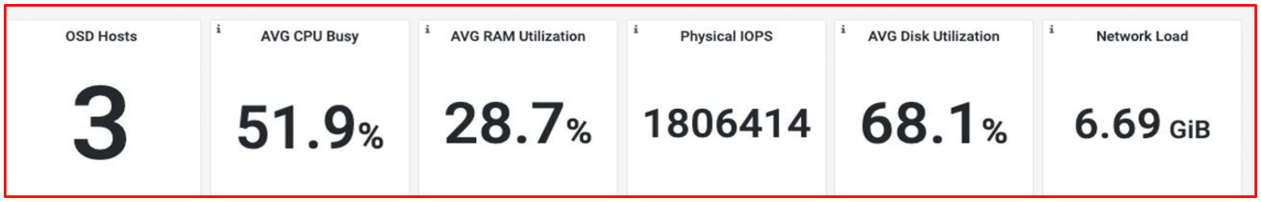
This approach is similar to the previous one, but instead of using new disks, we retain and reuse the existing NVMe disks that already have the data stored on them. It is important to verify that the disks are functional and in good condition before proceeding with this method. Check for any errors on PGs, and OSD logs before proceeding. *Note that this method is not for replacing a failed node, it is for migrating a node*.

9.2.1 Steps to be Followed

1. Set up an Ampere Arm node and install the operating system and Ceph prerequisites.
2. Disable backfilling in the cluster.
3. Shut down the Ceph service on the x86 node and verify that all Ceph services and containers are down.
4. Remove the disks from the x86 node and install them on the Ampere node.
5. Verify that the disks and LVMs are available and reboot the node if necessary.
6. Take a backup of the Ceph configuration on the x86 node, including /var/lib/ceph, /etc/ceph, and /etc/systemd/system files, maintaining the symbolic links.
7. Extract the Ceph configuration from the backup and apply it to the Ampere node.
8. Reconfigure the IP address and hostname on the Ampere node to match the x86 node.
9. Add the host with Ceph Orchestrator and start the ceph.target service on the Ampere node.
10. Re-enable the backfill parameter.

We observed the following from Ceph dashboard during the migration process (see *Figure 14*).

Figure 14: Dashboard Snapshot for Node Swap-Out and Disk Reuse

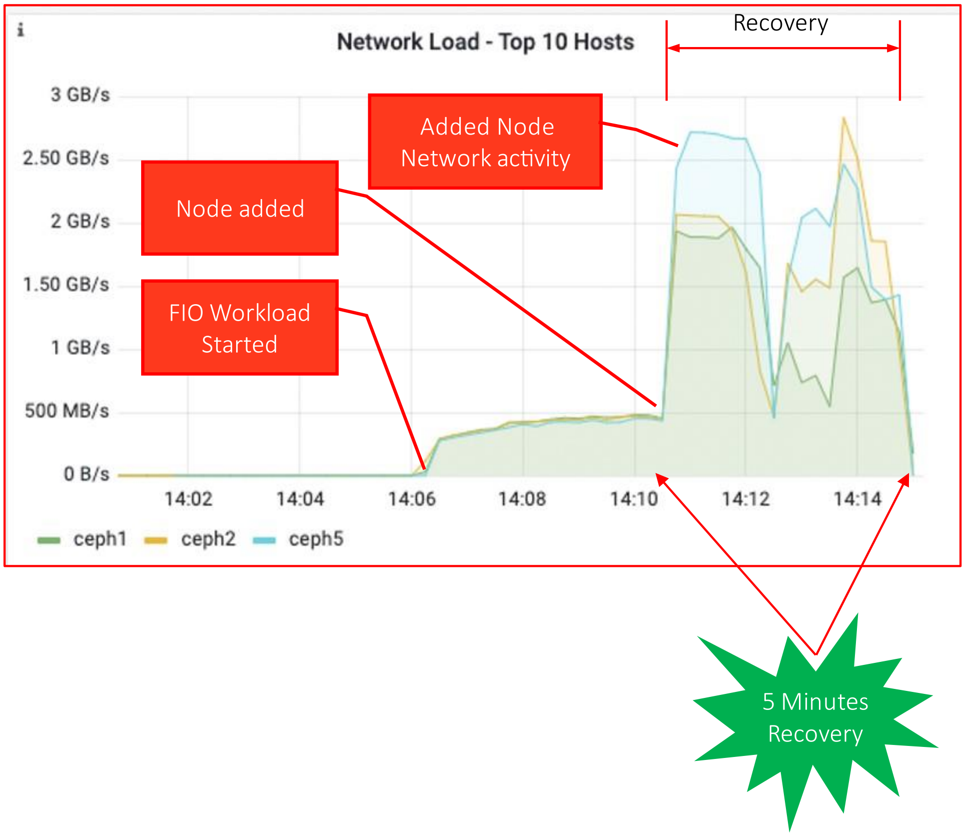


OSD Hosts	AVG CPU Busy	AVG RAM Utilization	Physical IOPS	AVG Disk Utilization	Network Load
3	51.9%	28.7%	1806414	68.1%	6.69 GiB

The system was performing around 1.8 million read and write IOPS in the backend, and the network was close to 6.7GB .

The follow graph captures our findings during the process (see *Figure 15*):

Figure 15: Read and Write IOPS During Node Swap-Out and Disk Reuse



Recovery

Added Node Network activity

Node added

FIO Workload Started

5 Minutes Recovery

9.2.2 Observations

- The backfill parameter was enabled after starting the FIO workload.
- System fully recovered to a healthy state in 5 minutes.

10. Ceph Tracker Issues

The following issues were discovered while running the tests on the test bed. Ampere Computing is following up these issues.

1. Crc32 issues on Arm having 12% overhead: https://tracker.ceph.com/issues/58352
2. Ceph HAProxy is missing Arm containers: https://tracker.ceph.com/issues/58367

11. Conclusion

The reference architecture and the solution presented in this paper is focused on running Ceph on a multi-node cluster with Ampere Arm processors. The community version of Ceph (Quincy) was run on Ampere processors and no issues were discovered in the test cases attempted. Apart from delivering good performance, we observed exceptional power savings while running Ceph. Further, customers can transition their existing Ceph clusters running x86 systems to Ampere-based systems without encountering major downtime disruptions.

12. Appendix

12.1 BIOS

BIOS: Advanced->ACPI Settings	Enable Max Performance
BIOS: Chipset->CPU Configuration->ANC mode	Monolithic(default)
BIOS: Chipset->Memory Configuration	Memory Speed 3200MT/s
IOS: SLC policy->Chipset->CPU Configuration ->SLC Replacement Policy	Enhanced Least Recently Used

12.2 Linux

```
CPU scaling governor – Performance
Transparent Huge Pages – Disabled
NVME read_ahead_kb – 2048 # Only for object storage tests
Swap – Disabled
Update /etc/default/grub with GRUB_CMDLINE_LINUX="iommu.passthrough=1" and update-grub2
```

12.3 /etc/sysctl.conf

```
net.ipv4.conf.default.rp_filter = 1
net.ipv4.tcp_timestamps = 0
net.ipv4.tcp_sack = 1
net.core.rmem_default = 33554431
net.core.wmem_default = 33554432
vm.swappiness = 0
vm.overcommit_memory = 0
net.core.somaxconn = 1024
net.core.netdev_max_backlog = 50000
net.ipv4.tcp_max_syn_backlog = 30000
net.ipv4.tcp_max_tw_buckets = 2000000
net.ipv4.tcp_tw_reuse = 1
net.ipv4.tcp_fin_timeout = 10
net.ipv4.tcp_slow_start_after_idle = 0
kernel.pid_max = 4194303
fs.aio-max-nr = 1048576
net.ipv4.ip_local_port_range = 8192 65535
fs.file-max=6553600
net.nf_conntrack_max=524288
net.netfilter.nf_conntrack_max=524288
```

12.4 /etc/security/limits.conf

```
* soft nofile 327680
* hard nofile 327680
```

12.5 Ceph

```
osd_pool_default_pg_num = 4096

log_to_syslog = true

mon_clock_drift_allowed = 0.15

mon_clock_drift_warn_backoff = 30

osd_pool_default_pg_num = 4096

bluestore_throttle_bytes = 268435456

bluestore_throttle_deferred_bytes = 536870912

bluestore_rocksdb_options =
compression=kNoCompression,max_write_buffer_number=128,min_write_buffer_number_to_merge
=16,compaction_style=kCompactionStyleLevel,write_buffer_size=8388608,max_background_job
s=4,level0_file_num_compaction_trigger=8,max_bytes_for_level_base=1073741824,max_bytes_
for_level_multiplier=8,compaction_readahead_size=2MB,max_total_wal_size=1073741824,writ
able_file_max_buffer_size=0

bluestore_cache_autotune = false

bluestore_cache_size_ssd = 25769803776

bluestore_cache_kv_ratio = 0.2

bluestore_cache_meta_ratio = 0.8

bluefs_buffered_io = false

[osd]

osd_client_message_cap = 1024

[client]

rbd_cache = false
```

12.6 FIO Commands

Single FIO writes/reads with synch

Write Tests

```
fio --filename=/dev/rbd0 --direct=1 -sync=1 --rw=randwrite --bs=4k --numjobs=1 --
iodepth=xxx --ramp_time=60 --runtime=300 --ioengine=libaio --size=200G --time_based --
group_reporting --name=cinder-test --eta-newline=5s
```

Read Tests

```
fio --filename=/dev/rbd0 --direct=1 -sync=1 --rw=randread --bs=4k --numjobs=1 --
iodepth=xxx --ramp_time=60 --runtime=300 --ioengine=libaio --size=200G --time_based --
group_reporting --name=cinder-test --eta-newline=5s
```

Multiple FIOs from multiple clients and RBDs

Write Tests

```
fio --direct=1  --rw=randwrite --bs=4k --numjobs=1 --iodepth=8 --ramp_time=60 --
runtime=600 --ioengine=rbd --pool=rbd --size=200G --time_based --group_reporting --
numa_cpu_nodes=0 --numa_mem_policy=bind:0 --name=cinder-test$i --rbdname=rbd$i --
output-format=terse --output=rbd${i}_writes.lst
```

Read Tests

```
fio --direct=1  --rw=randread --bs=4k --numjobs=1 --iodepth=16 --runtime=600 --
ioengine=rbd --pool=rbd --size=200G --time_based --group_reporting --numa_cpu_nodes=0 -
-numa_mem_policy=bind:0 --name=cinder-test$i --rbdname=rbd$i --output-format=terse --
output=rbd${i}_reads.lst &
```

12.7 COSbench – Workload-config.xml

`Snippet from conf/workload-config.xml for 10M puts`

```
    <workstage name="prepare">

      <work type="prepare" workers="100"
config="cprefix=s3testceph;containers=r(1,1);oprefix=run-
10m;objects=r(1,2000);sizes=c(10)MB" />

    </workstage>

    <workstage name="main" clousuredelay="60">

      <work name="main" workers="100" runtime="300" rampup="60">

        <operation type="write" ratio="100"
config="cprefix=s3testceph;containers=u(1,1);objects=r(1,2000);oprefix=run-
10m;sizes=c(10)MB" />

      </work>

    </workstage>

    <workstage name="cleanup">

      <work type="cleanup" workers="100"
config="cprefix=s3testceph;containers=r(1,1);objects=r(1,2000)" />

    </workstage>

    <workstage name="dispose">

      <work type="dispose" workers="1" config="cprefix=s3testceph;containers=r(1,1)" />

    </workstage>
```

Document Revision History

ISSUE	DATE	DESCRIPTION
1.00	February 22, 2023	Initial release.

February 22, 2023

Ampere Computing reserves the right to change or discontinue this product without notice.

While the information contained herein is believed to be accurate, such information is preliminary, and should not be relied upon for accuracy or completeness, and no representations or warranties of accuracy or completeness are made.

The information contained in this document is subject to change or withdrawal at any time without notice and is being provided on an "AS IS" basis without warranty or indemnity of any kind, whether express or implied, including without limitation, the implied warranties of non-infringement, merchantability, or fitness for a particular purpose.

Any products, services, or programs discussed in this document are sold or licensed under Ampere Computing's standard terms and conditions, copies of which may be obtained from your local Ampere Computing representative. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Ampere Computing or third parties.

Without limiting the generality of the foregoing, any performance data contained in this document was determined in a specific or controlled environment and not submitted to any formal Ampere Computing test. Therefore, the results obtained in other operating environments may vary significantly. Under no circumstances will Ampere Computing be liable for any damages whatsoever arising out of or resulting from any use of the document or the information contained herein.

Ampere Computing
4655 Great America Parkway, Santa Clara, CA 95054
Phone: (669) 770-3700
https://www.amperecomputing.com

Ampere Computing reserves the right to make changes to its products, its datasheets, or related documentation, without notice and warrants its products solely pursuant to its terms and conditions of sale, only to substantially comply with the latest available datasheet.

Ampere, Ampere Computing, the Ampere Computing and 'A' logos, and Altra are registered trademarks of Ampere Computing.

Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All other trademarks are the property of their respective holders.