WHITE PAPER

HIGH-PERFORMANCE ALL FLASH NVME CEPH CLUSTER ON SUPERMICRO X12 BIGTWIN® PLATFORM

Optimized Ceph cluster block storage performance with Supermicro BigTwin® 2U 2-Node servers and Upstream Ceph Storage with 3rd Gen Intel® Xeon® Scalable Processors on openSuse 15.2 to provide the highest performance architecture.

TABLE OF CONTENTS

Executive Summary

Enterprise storage infrastructure and related technologies continue to evolve year after year. In particular, as IoT, 5G, AI, and ML technologies are gaining attention, the demand for Software-defined storage (SDS) solutions based on clustered storage servers is also increasing; Ceph is a leading SDS solution that enables high performing workloads to run efficiently. The high throughput and low latency features of modern storage devices such as PCI-E 4.0 NVMe, Optane, and a variety of PCI-E network cards are essential factors that improve the overall performance of Ceph clusters. In addition, adopting a Ceph cluster that utilizes NVMe Solid State Disks (SSD) maximizes the overall application performance. Supermicro experts designed and assembled a Ceph cluster and then conducted various tests to validate these configurations. The clusters used in the benchmarks are based on the Supermicro BigTwin® SYS-220BT-DNTR, an all-flash storage server with the 3rd Gen Intel® Xeon® Scalable Processors with all-flash NVMe SSDs.

Mission Statement

SUPERMICRO

Supermicro (Nasdaq: SMCI), the leading innovator in high-performance, high-efficiency server and storage technology is a premier provider of advanced server Building Block Solutions® for Enterprise Data Center, Cloud Computing, Artificial Intelligence, and Edge Computing Systems worldwide. Supermicro is committed to protecting the environment through its "We Keep IT Green®" initiative and provides customers with the most energy-efficient, environmentally-friendly solutions available on the market.

This document will provide insights in recognizing the advantages and power of using Ceph storage with Supermicro hardware. Some of the advantages of this solution include a reduction in TCO and enhancements to performance and serviceability. The intended audience for this is IT Infrastructure and DevOps personnel to work towards a solution for their storage needs. This solution can broadly reach industry verticals, from hyperscalers to traditional IT shops.

Introduction to Ceph Community Edition

Ceph community edition uses cephadm to deploy and manage a Ceph cluster by connecting the manager daemon to hosts via SSH. The manager daemon is able to add, remove, and update Ceph containers. cephadm does not rely on external configuration tools such as Ansible, Rook, or Salt. Instead, it manages the entire lifecycle of a Ceph cluster. This lifecycle starts with the bootstrapping process when cephadm creates a tiny Ceph cluster on a single node. This cluster consists of one monitor and one manager. It then uses the orchestration interface ("day 2" commands) to expand the cluster, adding all hosts and provisioning all Ceph daemons and services. Management of this lifecycle can be performed either via the Ceph command-line interface (CLI) or via the dashboard (GUI).

All Ceph Storage Cluster deployments begin with setting up each Ceph Node, your network, and the Ceph Storage Cluster:

- Providing Ceph Object Storage and/or Ceph Block Device services to Cloud Platforms
- Deploying a Ceph File System
- Using Ceph for any other purpose

Ceph Community Edition uses the following components to form a Ceph Cluster:

Monitors: A Ceph Monitor (ceph-mon) maintains maps of the cluster state, including the monitor map, manager map, the OSD map, the MDS map, and the CRUSH map. These maps are critical cluster states required for Ceph daemons to coordinate with each other. Monitors are also responsible for managing authentication between daemons and clients. At least three monitor nodes are generally needed for redundancy and high availability.

Managers: A Ceph Manager daemon (ceph-mgr) is responsible for keeping track of runtime metrics and the current state of the Ceph cluster, including storage utilization, current performance metrics, and system load. The Ceph Manager daemons also host python-based modules to manage and expose Ceph cluster information, including a web-based Ceph Dashboard and REST API. At least two manager nodes are typically required for high availability.

Ceph OSDs: A Ceph OSD (object storage daemon, ceph-osd) stores data, handles data replication, recovery, rebalancing, and provides monitoring information to Ceph Monitors and Managers by checking other Ceph OSD Daemons for a heartbeat. At least three Ceph OSD nodes are normally required for redundancy and high availability.

MDSs: A Ceph Metadata Server (MDS, ceph-mds) stores metadata on behalf of the Ceph File System (i.e., Ceph Block Devices and Ceph Object Storage do not use MDS). Ceph Metadata Servers allow POSIX file system users to execute basic commands (like ls, find, etc.) without placing an enormous burden on the Ceph Storage Cluster.

CRUSH: Ceph stores data as objects within logical storage pools. Using the CRUSH algorithm, Ceph calculates which placement group should contain the object and further calculates which Ceph OSD Daemon should store the placement group. Thus, the CRUSH algorithm enables the Ceph Storage Cluster to scale, rebalance, and recover dynamically.

Ceph Block Device

A block is a sequence of bytes (often 512). Block-based storage interfaces are a mature and common way to store data on media, including HDDs, SSDs, CDs, floppy disks, and even tape. The ubiquity of block device interfaces is perfect for interacting with mass data storage, including Ceph.

Ceph block devices are thin-provisioned, resizable, and store data striped over multiple OSDs. Ceph block devices leverage RADOS capabilities, including snapshots, replication, and strong consistency. Ceph block storage clients communicate with Ceph clusters through kernel modules or the librbd library.

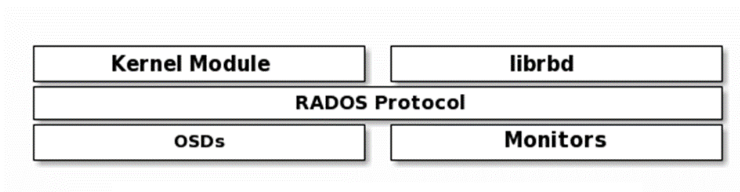

Figure 1 – Ceph Block Device

Ceph's block devices deliver high performance with vast scalability to kernel modules or KVMs such as QEMU and cloud-based computing systems like OpenStack and CloudStack that rely on libvirt and QEMU to integrate with Ceph block devices. Specifically, organizations that have chosen a private or hybrid cloud model can provide NVMe-backed Ceph RADOS Block Device (RBD) storage at a price point that is even more favorable than public cloud offerings while retaining essential performance characteristics. You can simultaneously use the same cluster to operate the Ceph RADOS Gateway, the Ceph File System, and Ceph block devices.

Supermicro Benchmark Setup

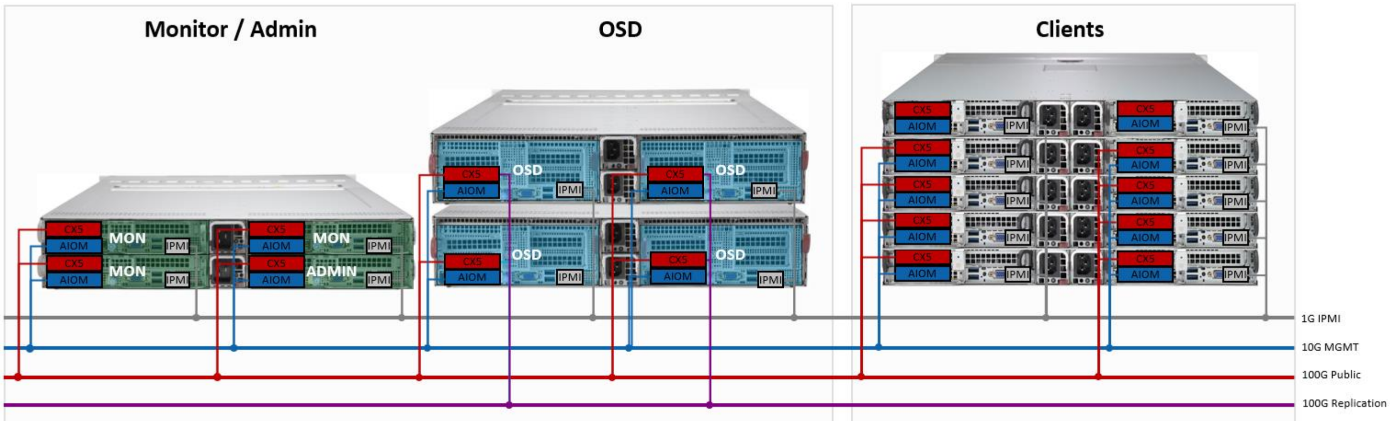

Figure 2 – Supermicro Ceph Benchmarking Setup

Supermicro has run several performance tests with the following setup. Figure 2 shows the Supermicro architecture with three monitor nodes, four Object Storage Daemon (OSD) nodes, and ten RADOS Block Devices (RBD) load-gen client nodes.

Supermicro Hardware BOMs and Software Specifics

The Ceph Storage cluster is deployed on the Supermicro BigTwin® two-node and four-node chassis containing 3rd Gen Intel® Xeon® Scalable Processors. The software versions used were Ceph Octopus on openSUSE Leap 15.2, and Flexible I/O Tester (FIO) 3.25.

OSD x4	
Type	**Description**
System	2x 2U 2-Node X12 BigTwin®
CPU	2x Intel® Xeon® Scalable @ 2.8GHz 32C / node
Memory	256GB DDR4-3200 RDIMMs / node
Boot Drives	Redundant NVMe M.2 Drives / node
Data Drives	12x NVMe PCI-E 4.0 Drives / node
AIOM	Dual-port 10GbE PCI-E 3.0 x8 NIC / node
AOC	Dual-port 100GbE PCI-E 4.0 x16 LP NIC / node

ADMIN x1 MON x3 RBD Clients x10	
Type	**Description**
System	4x 2U 4-Node X12 BigTwin®
CPU	2x Intel® Xeon® Scalable @ 2.0GHz 28C / node
Memory	256GB DDR4-3200 RDIMMs / node
Boot Drives	Redundant NVMe M.2 Drives / node
AIOM	Dual-port 10GbE PCI-E 3.0 x8 NIC / node
AOC	Dual-port 100GbE PCI-E 4.0 x16 LP NIC / node

Table 1 – BigTwin® BOMs

Software	Component	Value
Ceph Community Edition	Version	Ceph Octopus 15.2.12
	Config	/etc/ceph/ceph.conf
	Deployment	cephadm and cephorch
	PG num	4096
	# of OSD Daemons/NVMe	2
	Pool size	2
FIO	Version	3.25
	IO Engine	RBD, libaio
	Ramp time	300 seconds
	Run time	600 seconds
	RDB Size	100 GB
	Block Size	4KB (Random), 128KB (Sequential)
	Read/write mix	70/30 Read/Write
	IO Depth	1-32 (Random) 32 (Sequential)
	Num jobs	1
	Num Of RBD clients	200

Table 2 - Ceph-FIO settings

X12 BigTwin® All-in-one Rackmount Platform for Cloud Data Centers

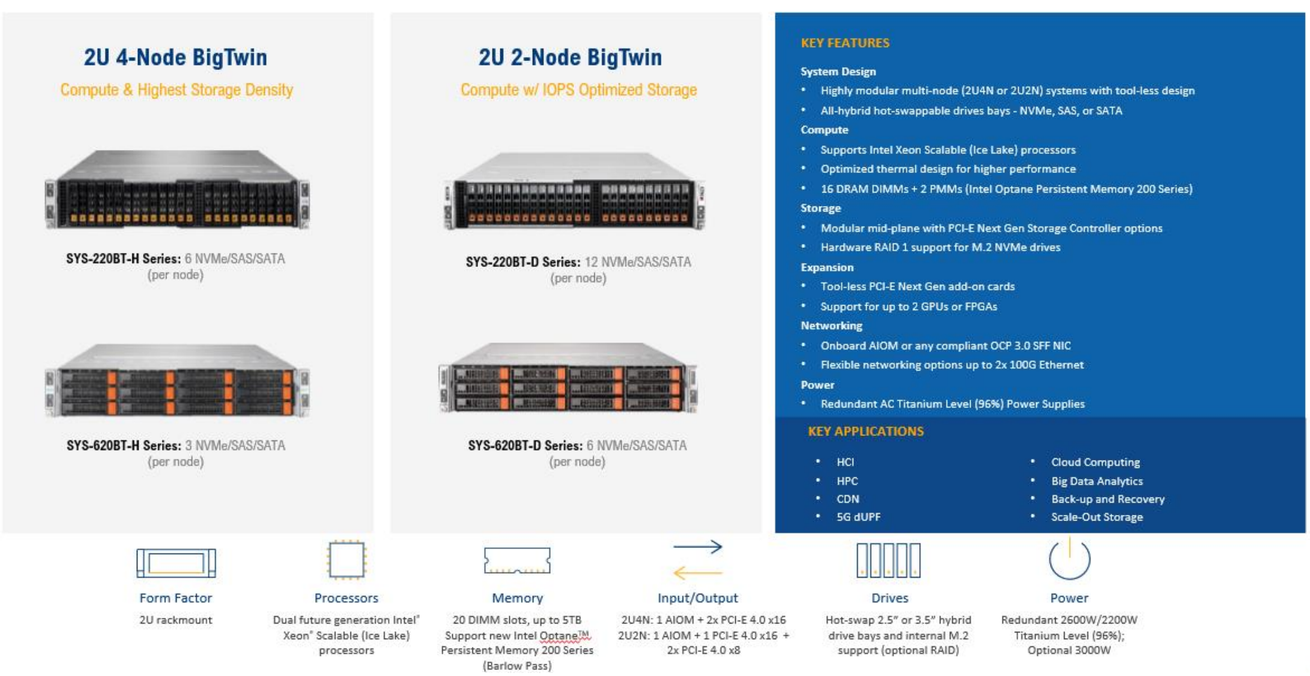

Figure 3 – BigTwin® Systems and Features

Supermicro's patented Twin Architecture is the foundation of the most energy efficient and advanced server platforms in HPC, Data Center, Cloud Computing, and Enterprise IT applications. These high performance, high density systems feature optimum airflow for energy efficient cooling, easy maintenance, and high availability with hot-swappable nodes and redundant power supply modules.

On SYS-220BT-DNTR, 12 NVMe PCI-E 4.0 drives deliver balanced performance with direct connections to the dual-processors in each node, as shown in Figure 4. The BigTwin® is one of the most forward-thinking and innovative server designs that reshaped our impressions on modular computing solutions by overcoming the traditional design compromises and obstacles.

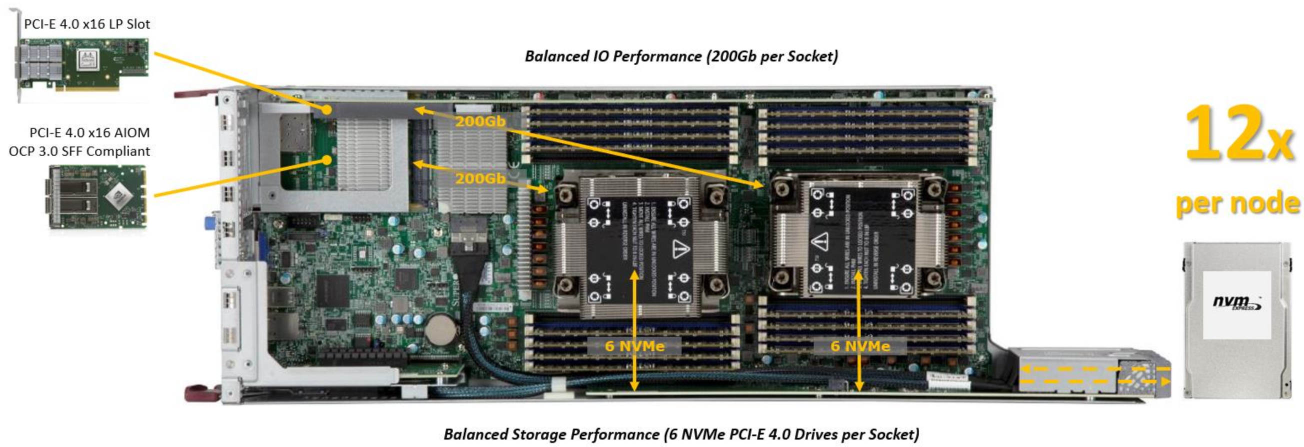

Figure 4 – Balanced Network and IO Performance

Baseline Test Results

This first test aims to measure the pure I/O performance of the storage for each node where the Ceph package is not installed. Then, the total performance of 48 NVMe PCI-E 4.0 drives across 2 enclosures with 4 nodes was measured by using the FIO (Flexible I/O tester) benchmark tool with the libaio IO engine. IOPS performance was evaluated for random IO workloads of a block IO size (4 KB). Sequential performance was also assessed for sequential IO workloads of a large block size (128 KB). The test was performed three times, and the results were averaged. The table below shows the baseline test results.

	4K Random Write			
	Avg. Throughput (KIOPS)	Avg. 99.99%th Latency (ms)	Avg. Latency (ms)	Avg. Throughput (GB/s)
OSD1	7948.33	8.12	0.39	30.30
OSD2	8523.00	1.96	0.36	32.53
OSD3	8656.33	1.92	0.35	33.00
OSD4	8664.33	1.88	0.35	33.07
	4K Random Read			
	Avg. Throughput (M IOPS)	Avg. 99.99%th Latency (ms)	Avg. Latency (ms)	Avg. Throughput (GB/s)
OSD1	13.00	6.59	0.24	49.53
OSD2	13.50	1.24	0.23	51.57
OSD3	13.33	0.90	0.23	50.97
OSD4	13.13	0.87	0.23	50.00

	128K Seq Write			
	Avg. Throughput (GB/s)	Avg. 99.99%th Latency (ms)	Avg. Latency (ms)	Avg. Throughput (K IOPS)
OSD1	43.77	21.98	1.07	358.00
OSD2	47.37	4.34	0.99	388.00
OSD3	47.50	4.29	0.99	389.00
OSD4	47.50	4.29	0.99	389.00
	128K Seq Read			
	Avg. Throughput (GB/s)	Avg. 99.99%th Latency (ms)	Avg. Latency (ms)	Avg. Throughput (K IOPS)
OSD1	75.97	5.32	0.62	622.33
OSD2	69.60	0.67	2.77	570.33
OSD3	77.90	0.60	1.01	638.00
OSD4	77.90	0.60	1.04	638.00

Table 3 -Baseline Results

Benchmark Configurations and Results

The following sections provide the results of synthetic benchmark performance for all-flash based Ceph clusters using Kioxia CM6 NVMe PCI-E 4.0 drives. The test was conducted in the RBD-based storage pool, which is the block storage component for Ceph. Workloads were generated using the FIO benchmark with ten client servers.

Before starting the test, Supermicro engineers created 200 RBD images that generated a total of 20 TB of data. Then, a 2x replication was applied, resulting in the total size of the data stored in the cluster being 40 TB.

• 10 Clients x 20 RBD images per client x 100 GB RBD image size = 20 TB (2x Replication: 20 TB x 2 = 40 TB)

A random test was created and run with a 4 KB IO workload and with the number of jobs equal to 1, and an iodepth of 1 - 32 per FIO instance. A sequential test was also created and run with a 128 KB IO workload with the number of jobs equal to 1 and an iodepth of 32 per FIO instance. We also measured latency variation across each test.

4 KB Random Read Workload

We measured the performance and latency of 4 KB random reads with increasing iodepth (1 - 32) on 200 clients. At an iodepth of 32, 4KB random read performance was measured at an average of 2.292 Million IOPS, with an average latency of 2.79ms and a tail latency (99.00th percentile latency) of 2.3ms. As the iodepth increased, IO performance and latency tended to increase. Maximum 2.294 Million IOPS was observed at an iodepth of 16 with an average latency of 1.39ms.

Iodepth	Bandwidth(GB/s)	K IOPS	avgLat(ms)	99.00pctLat(ms)	99.90pctLat(ms)	99.99pctLat(ms)
1	2.48	649.62	0.31	0.45	0.62	1.37
2	4.52	1184.36	0.34	0.53	0.71	1.89
4	7.03	1842.13	0.43	0.8	1.67	5
8	8.73	2289.17	0.7	2.02	40.53	87.12
16	8.75	2294.22	1.39	2.26	276.53	359.65
32	8.75	2292.46	2.79	2.3	760.13	846.07

Table 4 - 4K Random Reads

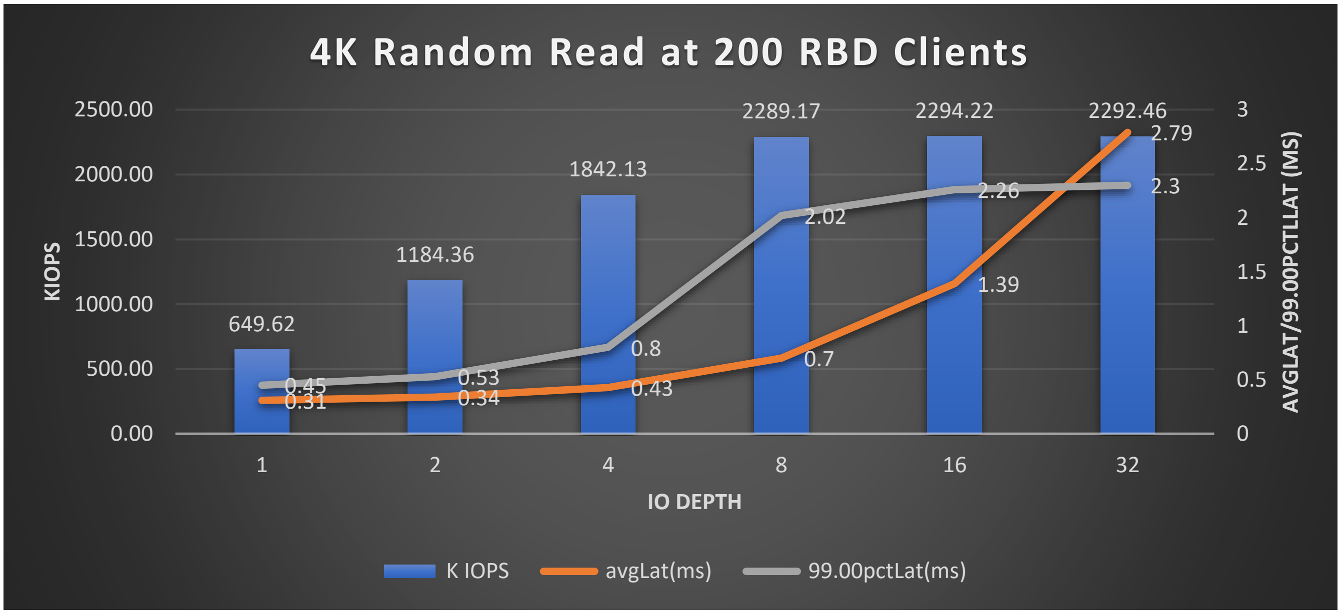

Figure 5 - 4K Random Reads at 200 RBD Clients

4 KB Random Write Workload

The benchmarks measured the performance and latency of 4 KB random writes with increasing iodepths (1 to 32) on 200 clients. At an iodepth of 32, 4 KB random write performance was measured at an average of 228.61 K IOPS, with an average latency of 28ms and an average tail latency (99.90th percentile latency) of 2988.71ms. As the iodepth increased, IO performance and latency tended to increase. Tail latency (99.90th percentile latency) increased significantly at iodepths of 32.

Iodepth	Bandwidth(GB/s)	K IOPS	avgLat(ms)	99.00pctLat(ms)	99.90pctLat(ms)	99.99pctLat(ms)
1	0.67	176.39	1.13	3.04	4.61	6.62
2	0.76	199.51	2	6.68	10.33	14.53
4	0.82	215.15	3.72	13.97	21.45	29.87
8	0.85	222.76	7.18	29.31	52.76	106.49
16	0.87	227.08	14.08	87.51	563.65	1362.81
32	0.87	228.61	28	313.97	2988.71	6102.15

Table 5 – 4K Random Writes

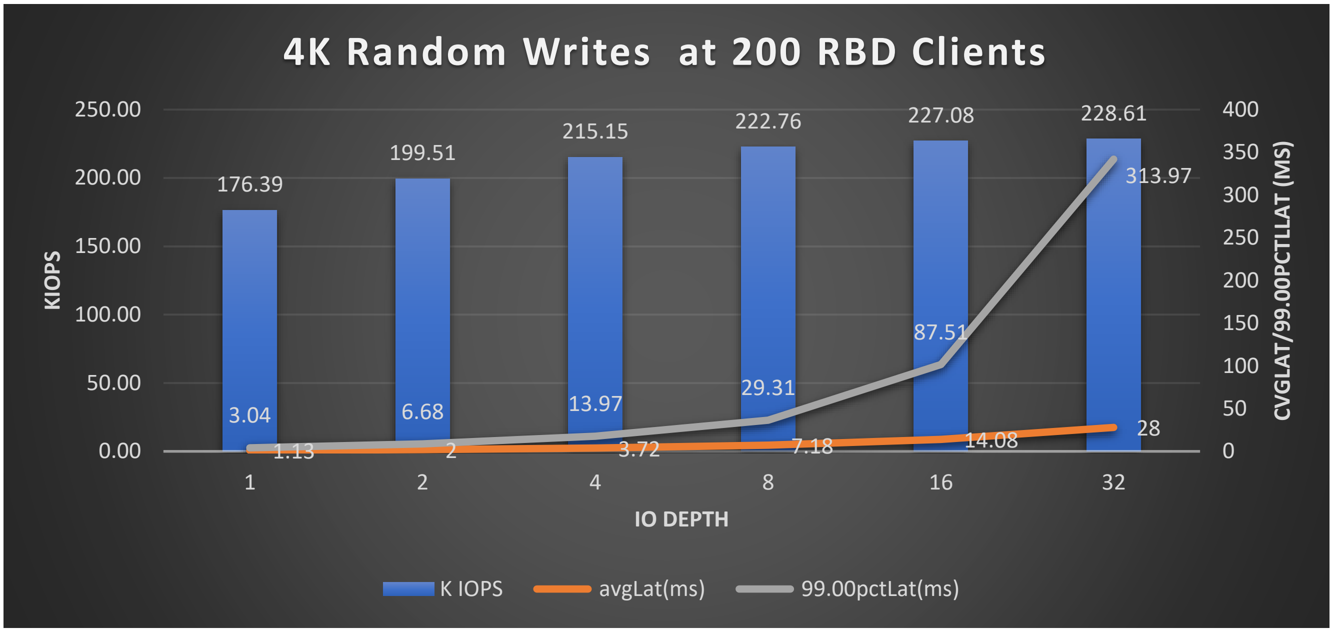

Figure 6 - 4K Random Writes at 200RBD Clients

4 KB Random ReadWrite 70/30

We measured the performance and latency of 4 KB random read/writes 70/30 with increasing iodepth at 200 clients. At an iodepth of 32, 4 KB random read/writes 70/30 performance was measured at an average of 531.2K IOPS, with an average latency of 12.06ms and an average tail latency (99.90th percentile latency) of 1585.35ms. As the iodepth increased, IO performance and latency tended to increase. Tail latency (99.90th percentile latency) increased significantly at the iodepth of 16 and higher.

iodepth	Bandwidth(GB/s)	K IOPS	avgLat(ms)	99.00pctLat(ms)	99.90pctLat(ms)	99.99pctLat(ms)
1	1.26	329.63	0.61	1.44	2.23	3.37
2	1.75	458.99	0.87	2.91	5.19	7.98
4	1.96	512.74	1.56	6.92	11.99	18.9
8	2.02	529.29	3.02	17.49	54.86	182.79
16	2.03	531.83	6.01	50.95	510.74	1173.94
32	2.03	531.20	12.06	163.58	1585.35	3077.25

Table 6 – 4K Random ReadWrites 70/30

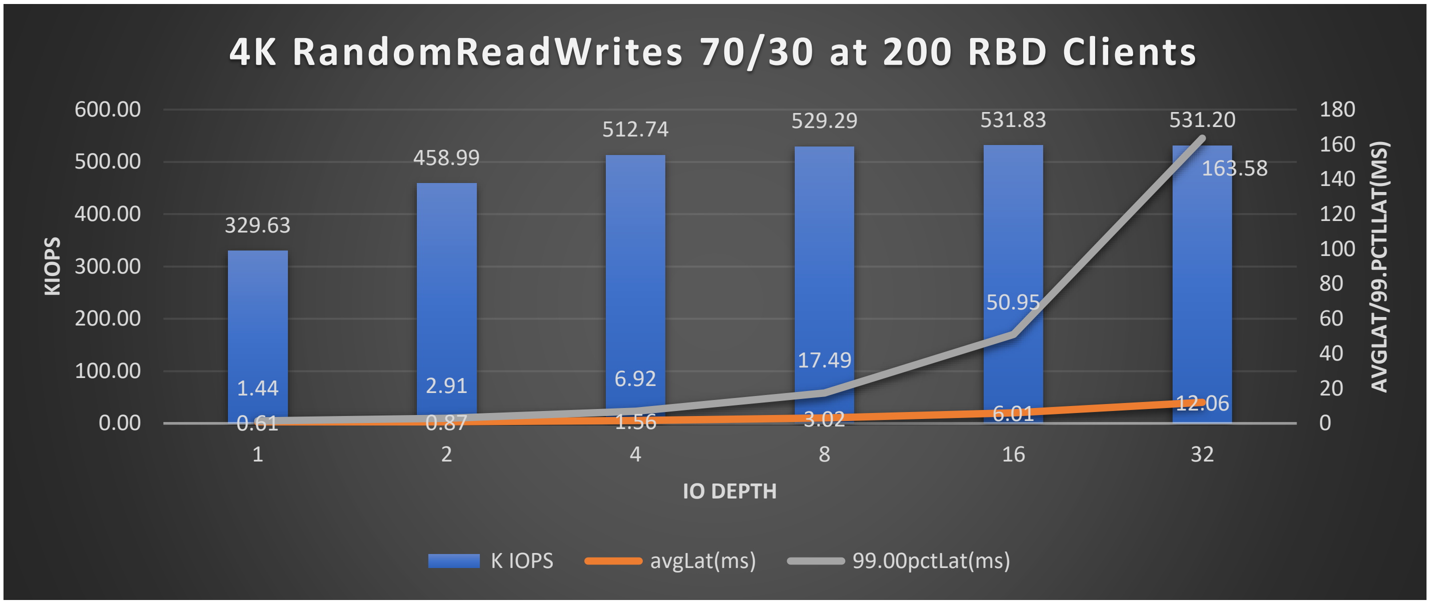

Figure 7 - 4K Random ReadWrites 70/30

Random Workloads CPU utilization

The CPU utilization has increased gradually as the iodepth increased for all three random workloads, as shown below in Figures 8, 9, 10. However, there's still headroom for more RBD clients, and even though IOPS started to taper off at 32 IO depth, the total aggregate throughput has room to scale with more clients.

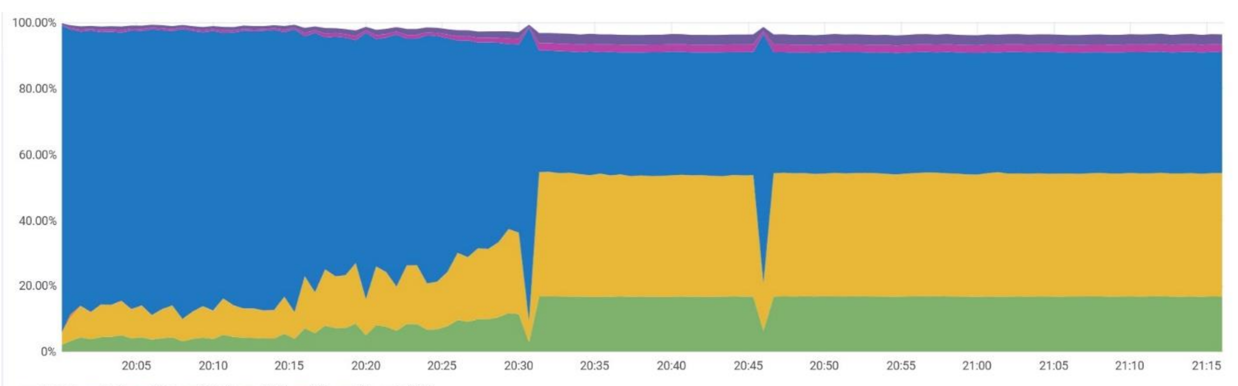

Figure 8 - OSD Nodes CPU Utilization-4K Random Reads Iodepth 1-32 Jobs

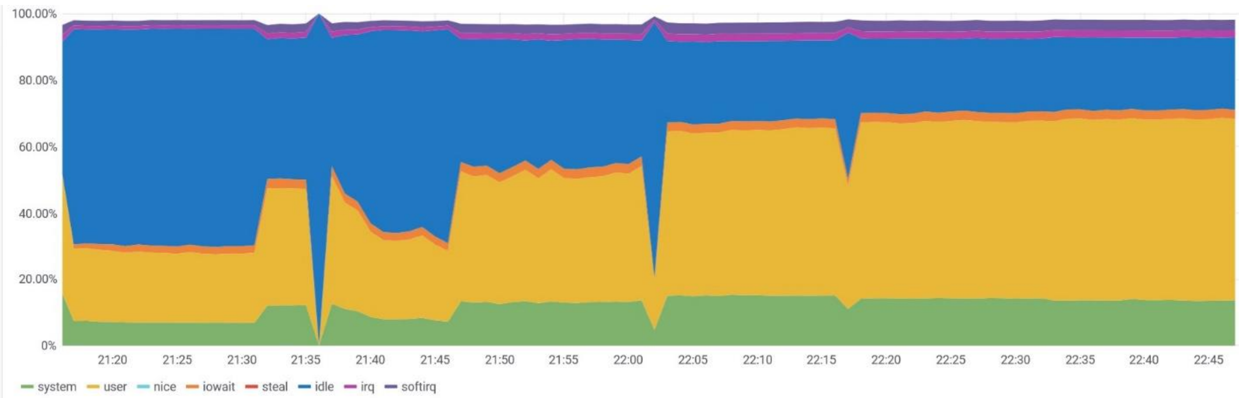

Figure 9 - 4K Random Writes OSD CPU Utilization Iodepth 1-32

Figure 10 - 4K Random ReadWrites 70/30 OSD CPU Utilization Iodepth 1-32

128 KB Sequential Read Workload

The maximum throughput of 41.98 GB/s was reached with 140 RBD clients for the 128 KB sequential reads. Latency increased steadily as the number of clients increased, while throughput remained relatively constant once the number of clients reached 80. The sequential reads workload caused the network bandwidth to approach its limits.

Procs	Bandwidth(GB/s)	K IOPS	avgLat(ms)
20	28.09	230.13	2.78
40	41.11	336.75	3.8
60	41.85	342.83	5.6
80	41.93	343.46	7.45
100	41.93	343.46	9.32
120	41.92	343.36	11.18
140	41.98	343.82	13.03
160	41.96	343.61	14.9
180	41.98	343.83	16.76
200	41.97	343.71	18.63

Table 7 – 128K Seq Reads at an Iodepth = 32

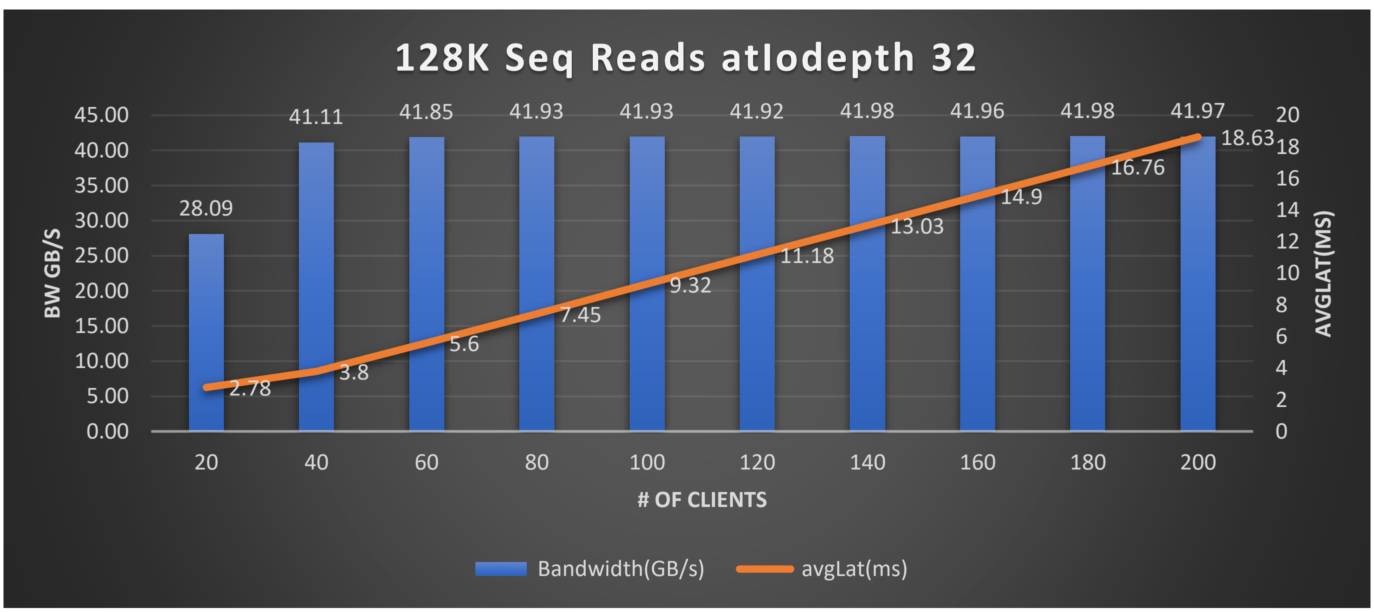

Figure 11 - 128K Seq Reads at Iodepth 32

128 KB Sequential Write Workload

The maximum sequential write throughput of 21.84 GB/s reached at 200 RBD clients for the 128 KB write workload. Latency increased steadily as the number of clients increased, while throughput remained relatively constant once the number of clients reached 120.

Procs	Bandwidth(GB/s)	K IOPS	avgLat(ms)
20	9.47	77.57	8.25
40	15.19	124.43	10.28
60	17.43	142.74	13.45
80	19.22	157.41	16.26
100	19.54	160.06	19.99
120	20.28	166.05	23.12
140	20.22	165.56	27.05
160	20.86	170.78	29.97
180	21.11	172.81	33.31
200	21.84	178.78	35.78

Table 8 -128K Seq Writes at Iodepth32

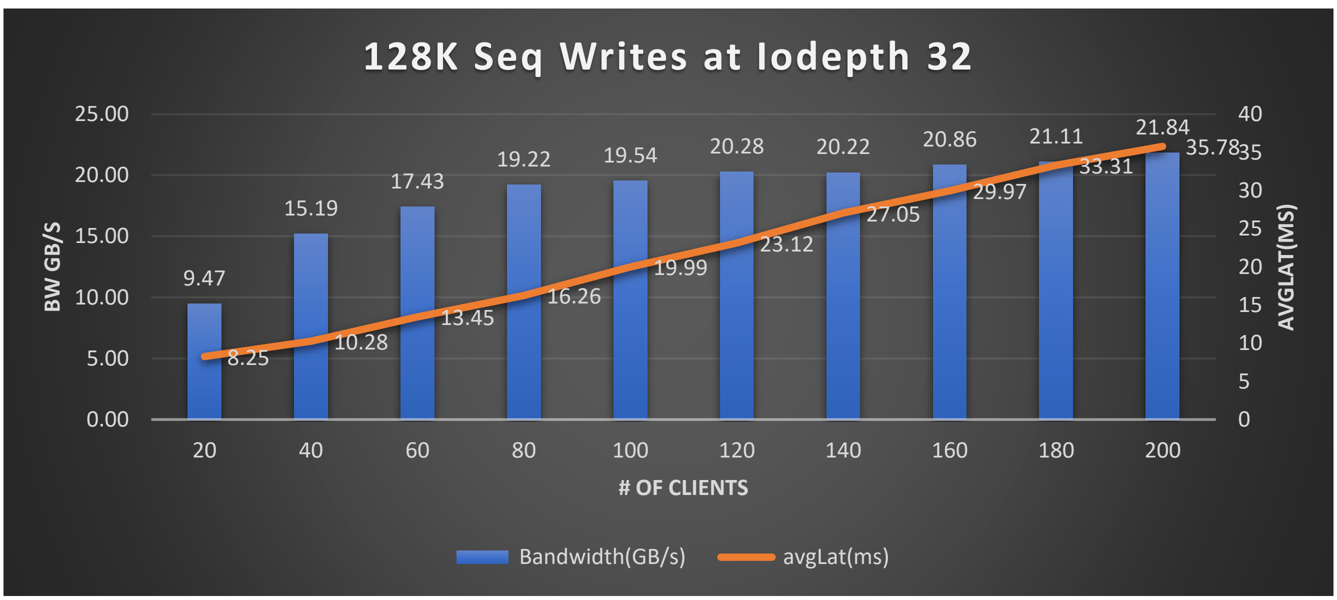

Figure 12 - 128K Seq Writes at 32 Iodepth

Ceph Configuration Files

ceph-conf.ini
 	
[global]
 	
mon_max_pg_per_osd=1000
 	
#mon_allow_pool_delete=true
 	
public network 10.5.5.0/24
 	
cluster network 192.168.5.0/24
 	
osd_max_pg_log_entries=10
 	
osd_min_pg_log_entries=10
 	
osd_pg_log_dups_tracked=10
 	
osd_pg_log_trim_min=10
 	
osd_enable_op_tracker=False
 	
max_open_files=500000
 	
bluefs_buffered_io=false
 	
rgw_list_buckets_max_chunk=999999
 	
rgw_override_bucket_index_max_shards=199
 	
rgw_dynamic_resharding=0
 	
rgw_bucket_index_max_aio=4096
 	
rgw_put_obj_max_window_size=134217728
 	
objecter_inflight_op_bytes=1048576000
 	
objecter_inflight_ops=102400
 	
ms_dispatch_throttle_bytes=1048576000
	ceph_spec.yml
	
	service_type: osd
	
	service_id: nvme_group
	
	placement:
	
	  host_pattern: 'osd[1-5]'
	
	data_devices:
	
	  model: 'KCM6XRUL3T84'
	
	osds_per_device: 2
	
	#db_devices:

rgw_obj_stripe_size=262144 rgw_max_chunk_size=262144 rbd_readahead_disable_after_bytes=0 rbd_readahead_max_bytes=4194304 bluestore_default_buffered_read=false mon_allow_pool_delete=true mutex_perf_counter=false throttler_perf_counter=false mutex_perf_counter=false throttler_perf_counter=false bluestore_cache_autotune=0 bluestore_rocksdb_options="compression=kNoCompression,max_write_buffer_number=32,min_write_buffer_number_to_merge=2,recycle_log_file_num=32,compaction_style=kCompactionStyleLevel,write_buffer_size=4MB,target_file_size_base=4MB,max_background_compactions=64,level0_file_num_compaction_trigger=16,level0_slowdown_writes_trigger=128,level0_stop_writes_trigger=256,max_bytes_for_level_base=512MB,compaction_threads=32,flusher_threads=8,compaction_readahead_size=2MB" bluestore_cache_meta_ratio=0.8 bluestore_cache_kv_ratio=0.2	

Figure 13 – Ceph Configuration Files

Summary of Ceph cluster RBD Storage Performance

Random Workload Pattern at 32 Iodepth	IOPS	Latency
4K 100% Random Reads	2.29mil	2.79 ms
4K 100% Random Writes	228.61K	28 ms
4K 70%/30% Read/Write Mix	531.23K	12.06
Seq Workload Pattern at 180 RBD Clients	**GB/S**	**Latency**
128K Seq Reads	41.97	18.63 ms
128K Seq Writes	21.84	35.78 ms

Table 9 - RBD Storage Performance

Conclusion

Supermicro BigTwin® servers are optimized for large cloud data center deployments and enterprise environments to deliver consistently high performance, making them an ideal solution for software-defined storage such as Ceph Storage. Supermicro has designed a performance-optimized, all-flash-based Ceph cluster using the 2U 2-Node BigTwin® for OSD nodes (SYS-220BT-DNTR) & 4-Node chassis MON nodes (SYS-220BT-HNTR), both use 3rd Gen Intel® Xeon® Scalable Processors 8362 CPUs, PCI-E 4.0 NVMe SSDs, and Ceph Storage-Octopus. This solution achieves over 2.29 million IOPS for the 4 KB random read workloads and is excellent sequential read throughput.

With a balanced CPU architecture, the Supermicro BigTwin® systems are optimized for scalable compute, database, tiered storage, and I/O intensive applications. In addition, with the latest U.2 NVMe PCI-E 4.0 drives, the CPU-to-drive ratio provides maximum balanced bandwidth while the All-Flash NVMe-based configurations deliver extreme storage performance with the highest IOPS per GB/system to help with data intensive workloads in a complex IT infrastructure.