TCO in Computational Storage - Compression

Jonmichael Hands, Chia Network

Tim Anderson, Scaleflux

SNIA Computational Storage Special Interest Group (SIG)

Table of Contents

List of Figures

Abstract

Computational storage (CS) transforms how we handle data, and this paper takes a deep dive into how compression impacts performance and Total Cost of Ownership (TCO) for storage systems. Computational storage brings compute closer to the data and accelerates common data processing workloads, including data reduction techniques like compression to free up valuable CPU resources. Compression impact on TCO is a huge lever. It can help us store more with the same space and reduce the total cost - but compression can slow things down and put more work on the CPU. We turn to computational storage to seamlessly offload the compression. We will look at a case study with Ceph, enabling compression on a single node cluster and comparing it with computational storage. We will use the SNIA Storage TCO model to show the impact of CS on storage costs.

Introduction to Computational Storage

Computational storage is all about doing more with less: less bandwidth, less system resource utilization, and less cost; while increasing performance, abilities, and impact. To accomplish this, computational storage improves the innate processing abilities of the storage device.

All SSDs have a microcontroller used for communication with the host CPU, management of data placement, monitoring and mitigating the health of the NAND, identifying and resolving errors, improving performance, security and encryption, and so much more. And with the ever-increasing PCIe generation performance bumps, these controllers continue to scale. But they still lack many computational abilities that can further enhance their benefit to applications and users in performance, drive capacity, security, and application acceleration to name a few.

Computational storage comes in a few different flavors: performing operations on data-in-flight before being written to or after being read from NAND, or executing compute operations on data at rest and sending only post-processed data to the host. For example, data-in-flight compute would be compression, encryption, and transcoding. And for compute operations on data at rest, examples are Key-Value SSDs or application specific in-situ processing.

In all cases, the goal is to increase performance/capacity/abilities while decreasing TCO. As storage demands continue to grow, computing closer to the data, inside the storage device, on dedicated hardware will enable us to keep up and continue to yield benefits from the immense storage pool available to us.

Successful broad adoption of computational storage requires clear technical benefits and easy integration. Enabling computational storage with drop-in compatibility with any system is requisite: no special adjacent hardware dependencies, no custom drivers, and no software changes. Performing compute within the block device completely self-contained is how this will be accomplished. The clearest application of such computational storage today is performing compression within the SSD on all data in flight. Compression, transparent to the host OS, software stack, and file system, improves performance, consistency, scalability, and return on investment.

Total Cost of Ownership

How does compression reduce TCO?

Compression reduces Total Cost of Ownership (TCO) in storage by increasing the effective capacity, denoted in “TBe” (terabytes effective). When data is compressed, more information can be stored in the same physical space, reducing the amount of hardware required. Higher compression ratio proportionally reduces the TCO $ / TBe, making it one of the largest impacts to data center storage TCO because it operates on the total number for both CapEx and OpEx!

How does compression in CS further reduce TCO?

Compression within computational storage systems provides a powerful mechanism for further reduction of TCO. At the system level, offloading data compression tasks from the host CPU to the storage hardware itself can reduce the need for high-end, high-core-count CPUs. This could lead to significant savings in hardware procurement and potentially reduce per-core application licensing costs.

Further reductions in TCO can be achieved through power savings. The offloading of compression tasks decreases CPU and memory power utilization. This process, in conjunction with a reduced number of drives in the system, decreases system volume and thus airflow, leading to lower cooling costs.

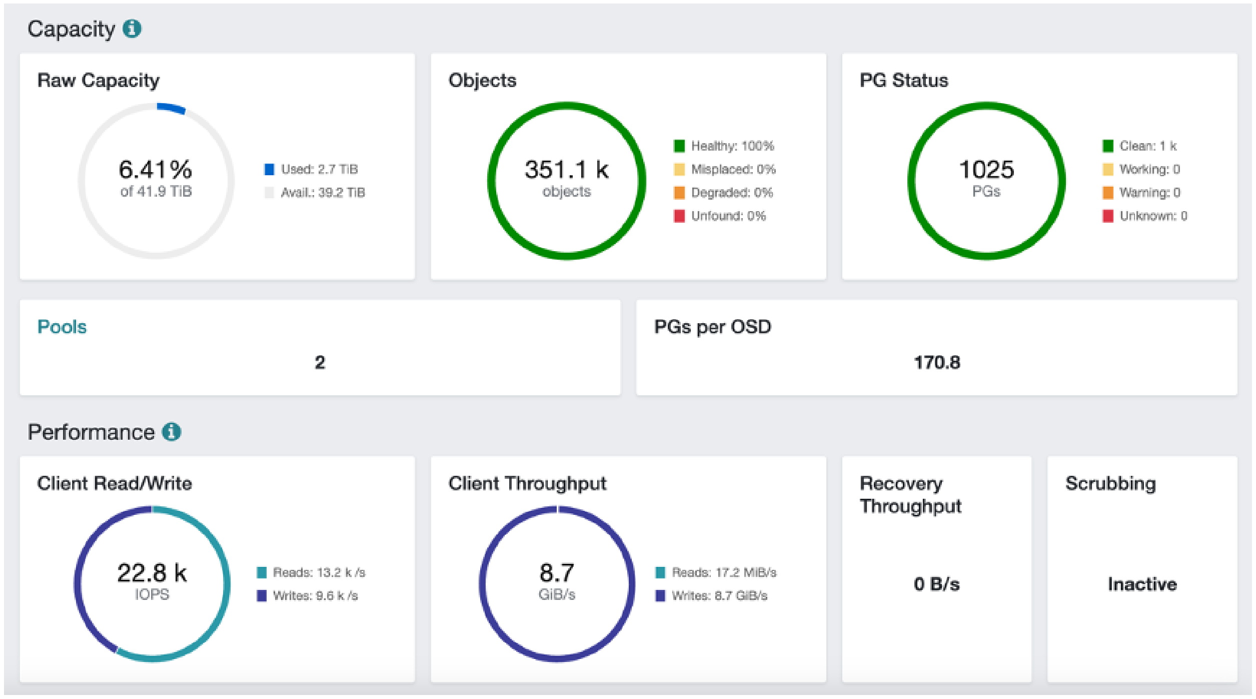
Computational storage can have an impact on endurance as well. When the host writes data to the drive, the computational storage device has to write less data to the NAND itself, reducing the number of NAND writes. Compression in CS makes it possible to have a number less than 1 for WAF (Write Amplification Factor)! Decreased WAF means higher performance and endurance over time.

Case Study - Computational Storage in Ceph

Compression in Ceph

Ceph is a powerful, scalable, and open-source distributed storage system that offers a multitude of storage capabilities in one unified system. Its object storage system, RADOS (Reliable Autonomic Distributed Object Store), provides the foundation for all of Ceph's storage capabilities, offering a highly available and fault-tolerant storage solution. Ceph stands out because it is designed to run on commodity hardware, making it cost-effective for businesses of all sizes. It provides excellent scalability, allowing organizations to simply add more hardware to scale out their storage infrastructure as needed. With its built-in redundancy and self-healing features, Ceph ensures data is always accessible and safe. Ceph has the ability to integrate with cloud platforms and support for various interfaces such as Block, Object, and File system make it a versatile solution for a wide range of applications. Ceph has various options for compression, including snappy, zlib, zstandard, and lz4. This versatility and cost-effectiveness make Ceph a compelling case study for computational storage.

Figure 1. Screenshot of the Ceph dashboard for the case study



Capacity

Raw Capacity
6.41% of 41.9 TiB
Used: 2.7 TiB
Avail.: 39.2 TiB

Objects
351.1 k objects
Healthy: 100%
Misplaced: 0%
Degraded: 0%
Unfound: 0%

PG Status
1025 PGs
Clean: 1 k
Working: 0
Warning: 0
Unknown: 0

Pools
2

PGs per OSD
170.8

Performance

Client Read/Write
22.8 k IOPS
Reads: 13.2 k /s
Writes: 9.6 k /s

Client Throughput
8.7 GiB/s
Reads: 17.2 MiB/s
Writes: 8.7 GiB/s

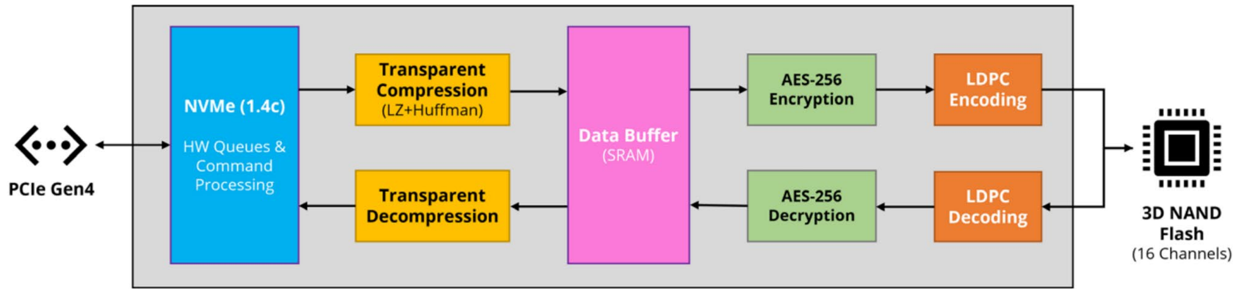
Recovery Throughput
0 B/s

Scrubbing
Inactive

Device Under Test - ScaleFlux CSD-3000

ScaleFlux Solid State Drives leverage a custom System-on-Chip (SoC) solution to deliver NVM Express features plus compute operations. This ARM-based SoC delivers key subsystems unique among NVMe Controllers. With eight A53 processors divided into two clusters, in-line computation can be performed on all IO with no impact to latency. The CSD-3000 utilizes a proprietary hardware accelerated lossless compression algorithm similar to DEFLATE (Lempel-Ziv + Huffman Coding), which is also used in the popular gzip and zlib compression formats. 4K chunks of data are compressed in real-time to variously sized smaller chucks. To efficiently manage these non-uniform compressed chunks, ScaleFlux implemented a novel Variable Length Flash Translation Layer, overcoming the limitation seen by other compression solutions that must zero pad the data to complete the minimum block size of 4K. By managing compression and decompression within the drive no custom drivers or software stack changes are needed. To capitalize on the capacity benefits, NVMe thin-provisioning practices can be used, thus allowing for up to 11.52TB of logical data to be stored on a ScaleFlux CSD-3000 with native capacity of 3.84TB.

Figure 2. Data Path in ScaleFlux CSD-3000


PCIe Gen4
NVMe (1.4c)
HW Queues & Command Processing
Transparent Compression (LZ+Huffman)
Transparent Decompression
Data Buffer (SRAM)
AES-256 Encryption
AES-256 Decryption
LDPC Encoding
LDPC Decoding
3D NAND Flash (16 Channels)

Test Case 1: Zstd

In Test Case 1, we enable compression on the pool to use zstandard with force to always enable.

Ceph Compression:

Mode: Force

Algorithm: zstd

Minimum blob size: default

Maximum blob size: default

Ratio: default

Test Case 2: Computational Storage Device

Compression in the software is disabled, leaving compressible data written directly to the drives, where the onboard compression engine will compress the data.

Ceph Compression:

Mode: Off

Algorithm: N/A

Minimum blob size: default

Maximum blob size: default

Ratio: default

Results

Figure 3. Test Case Results

Test Case	Bandwidth App (GB/s)	Bandwidth Disk (GB/s)	CPU Utilization	Compression Ratio
Zstd Write	6.3	5.6	93%	2.285
Zstd Read	21.1	9.2	87.8%	
CSD Write	8.3	16.79	66.5%	2
CSD Read	22.5	22.6	88.1%	

The bandwidth to the raw disks is increased significantly, which makes sense since in test case 1 the data is getting compressed before it gets written to disk. The CSD delivers 32% more write bandwidth from the application (this is before the 2x replication factor), significantly improving the performance of the cluster. For a sequential read, test case 2 delivers an additional 1.4 GB/s of bandwidth.

Figure 4. Bandwidth

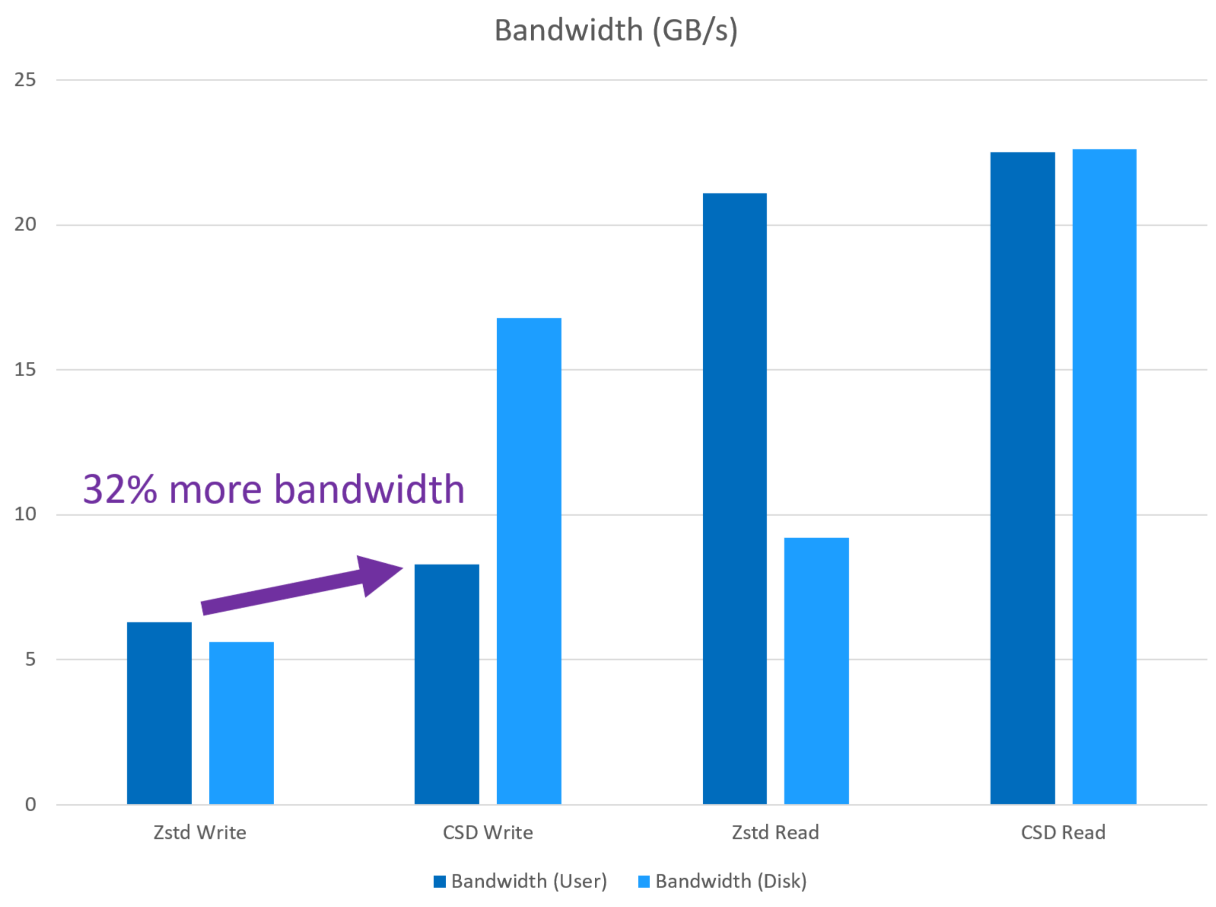
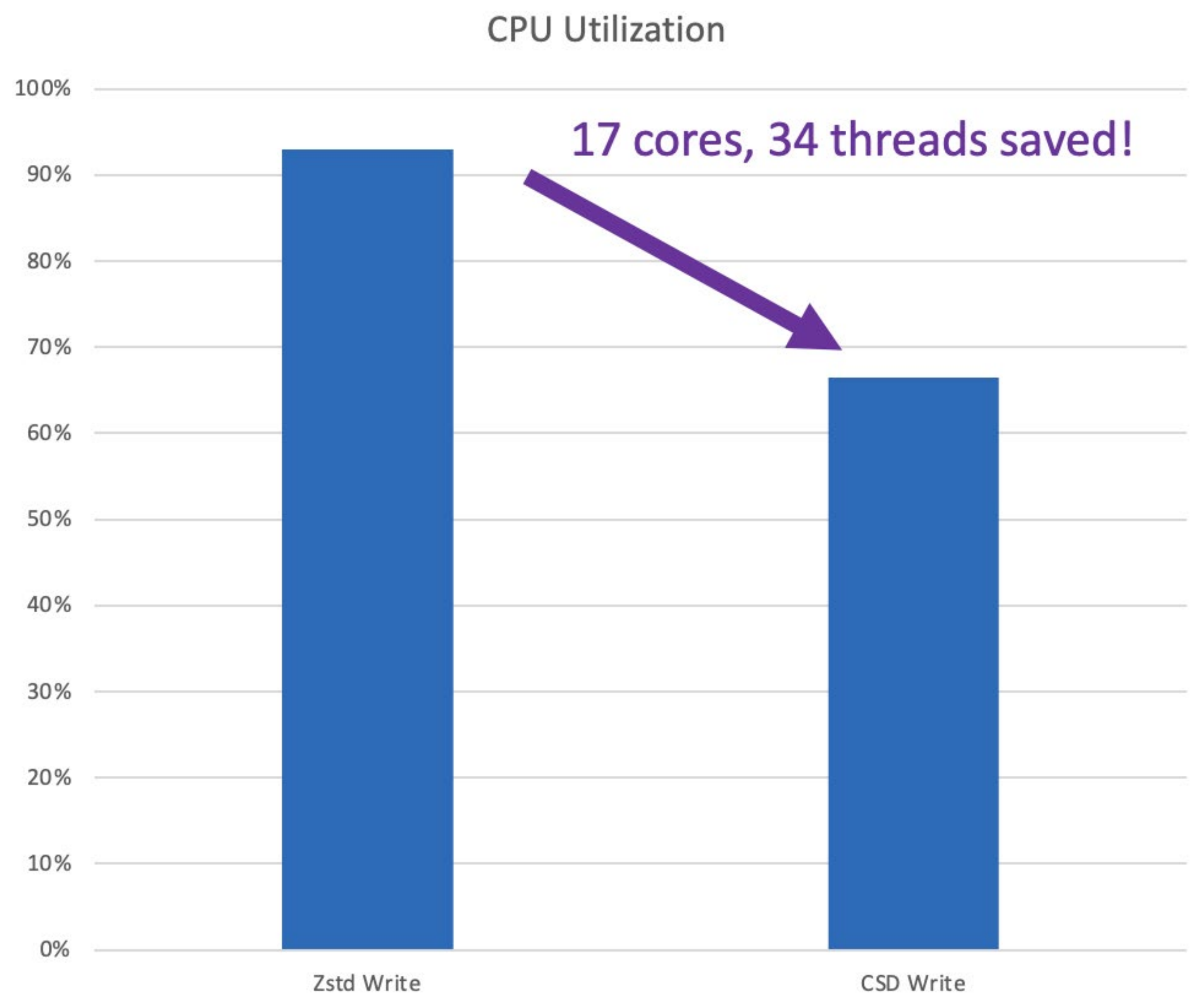


Figure 5. CPU Utilization



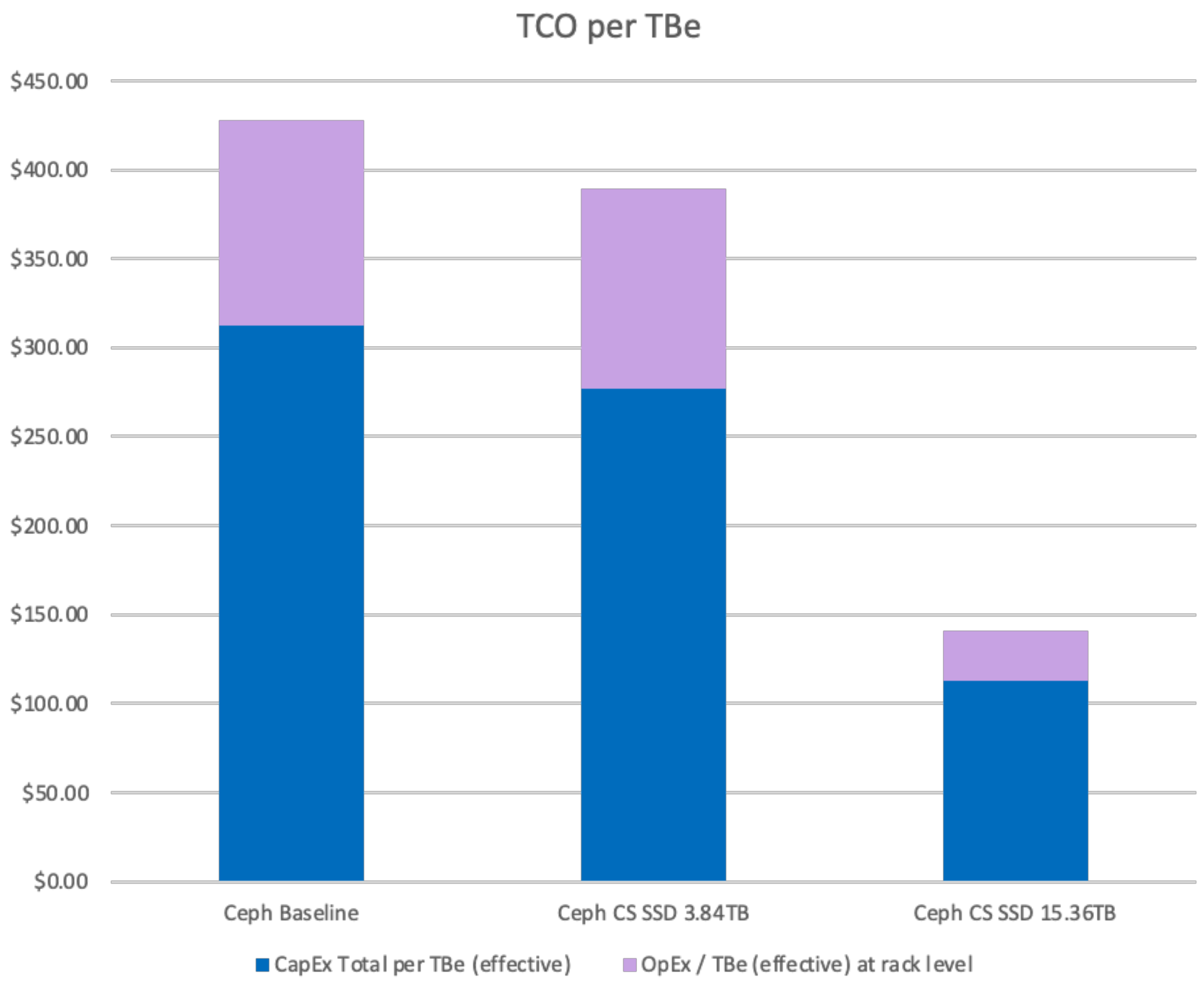
As predicted, the CSD solution requires less CPU cycles for the overall cluster io performance due to the CSD compressing and decompressing local to the device, rather than using host CPU cycles. The CSD saves 26.5% of the total CPU utilization, which on a dual 32 core system (128 threads) equates to 17 physical cores and 32 threads. This suggests that we could achieve the same cluster performance with two 24 core CPUs rather than two 32 cores, which would result in a server level decrease in CapEx.

TCO model results

In the TCO model, we model the server configuration with two CPU options.

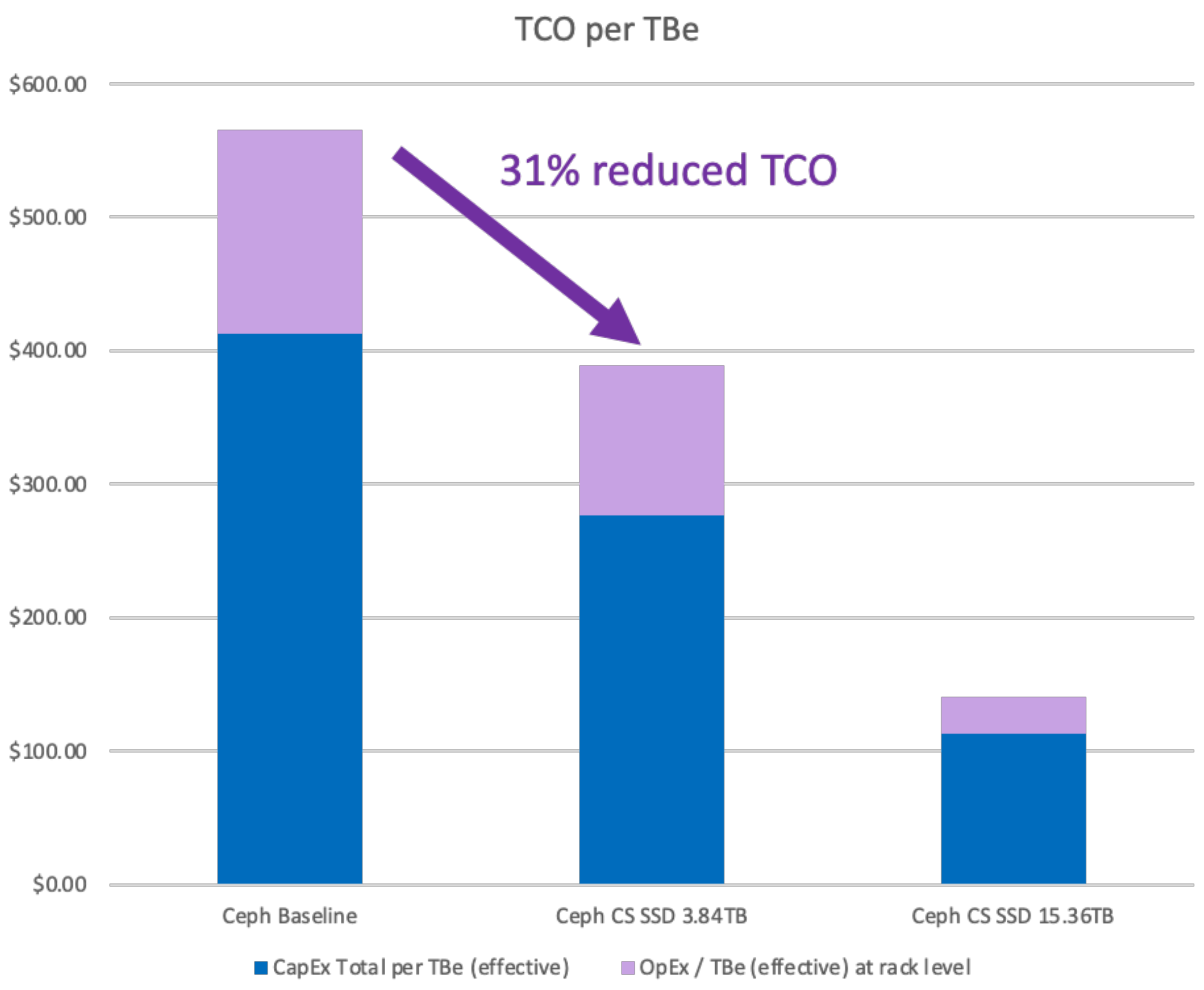
2x 32 core Intel Xeon processors with a recommended customer price of $2990 each vs two 24 core CPUs, with a recommended customer price of $1602. We use the original quote for the Supermicro server with the modified CPU prices to show a per-server cost reduction of $2776

Figure 6. TCO Model Output with only CapEx reduction from server and CPU power reduction



If we just include the CPU reduction from 32 core to 24 core, we observe a TCO reduction of 9%. If we include the performance increase (if we need to hit a bandwidth per node target), then the TCO savings increase to 31%!

Figure 7. TCO model output, Including performance multiplier of 1.32x from measured results



The test used 3.84TB drives, but as you can see from the TCO model output increasing the capacity per drive has a dramatic impact on TCO in this small 6 node cluster, due to the overhead of everything else in the rack. This is why high capacity NVMe SSDs much better solution when optimizing for TCO, if other performance requirements of the cluster are already met.

Limitations

This experiment did not push the capacity of the drives to their logical limit by creating a thin provisioned namespace. ScaleFlux has a nice setup guide for management at the drive level. The computational storage in this case required a bit more planning and testing upfront, and using the drive's tools to monitor the data reduction ratio.

Ceph is not meant to be run in a single cluster configuration, for data durability requirements to replicate data over many different physical servers. If we had more time, we would have liked to compare the 4-2 EC, and different replication methods (1x, 2x, 3x) for performance.

Other forms of hardware-accelerated compression reside outside the block device, e.g. accelerator card. Ceph can support Intel QAT, but a custom version of Ceph needs to be compiled and the cards need to be obtained. These were not included in this performance and cost comparison.

Discussion

The results support the original hypothesis that computational storage devices can reduce compute requirements for compression by offloading cpu cycles required. Even though a synthetic benchmark of FIO was run, the Ceph storage system still has to do the data protection and replication that showcases what real-world overhead durable storage systems have. This supports the industry trend toward computational storage and accelerators for tasks such as encryption, compression, and data protection. The TCO model shows that a reduction in CPU cost for the CapEx of the server results in modest TCO savings, where more aggressive TCO savings can be achieved with the use of higher capacity SSDs and baselining performance requirements of the storage system. A system administrator needs to determine the performance, network bandwidth, endurance, capacity, management and many other factors determining the purchasing requirements, the SNIA TCO model requires some understanding of the purchasing requirements of the customer to make an adequate comparison.

Appendix

System Configuration

System Config

- Single Node Cluster, Ice Lake Server
- Supermicro Ultra SuperServer SYS-120U-TNR
- 2x Intel Xeon Gold 6338 CPU
- 512GB DDR4 @3200MHz, 32x 16GB DIMMs
- 12x ScaleFlux CSD 3000 3.84TB

Software Config

- Ubuntu 23.04, Kernel 6.2.0-20-generic
- ceph version 17.2.6 quincy
  - Pool with 2x replication
  - 40x RBD of 1TB each
- fio-3.33
  - 128k sequential read, write, random read, QD 128
  - Buffer compress = 60 (average 2:1 compression ratio)

Ceph Config

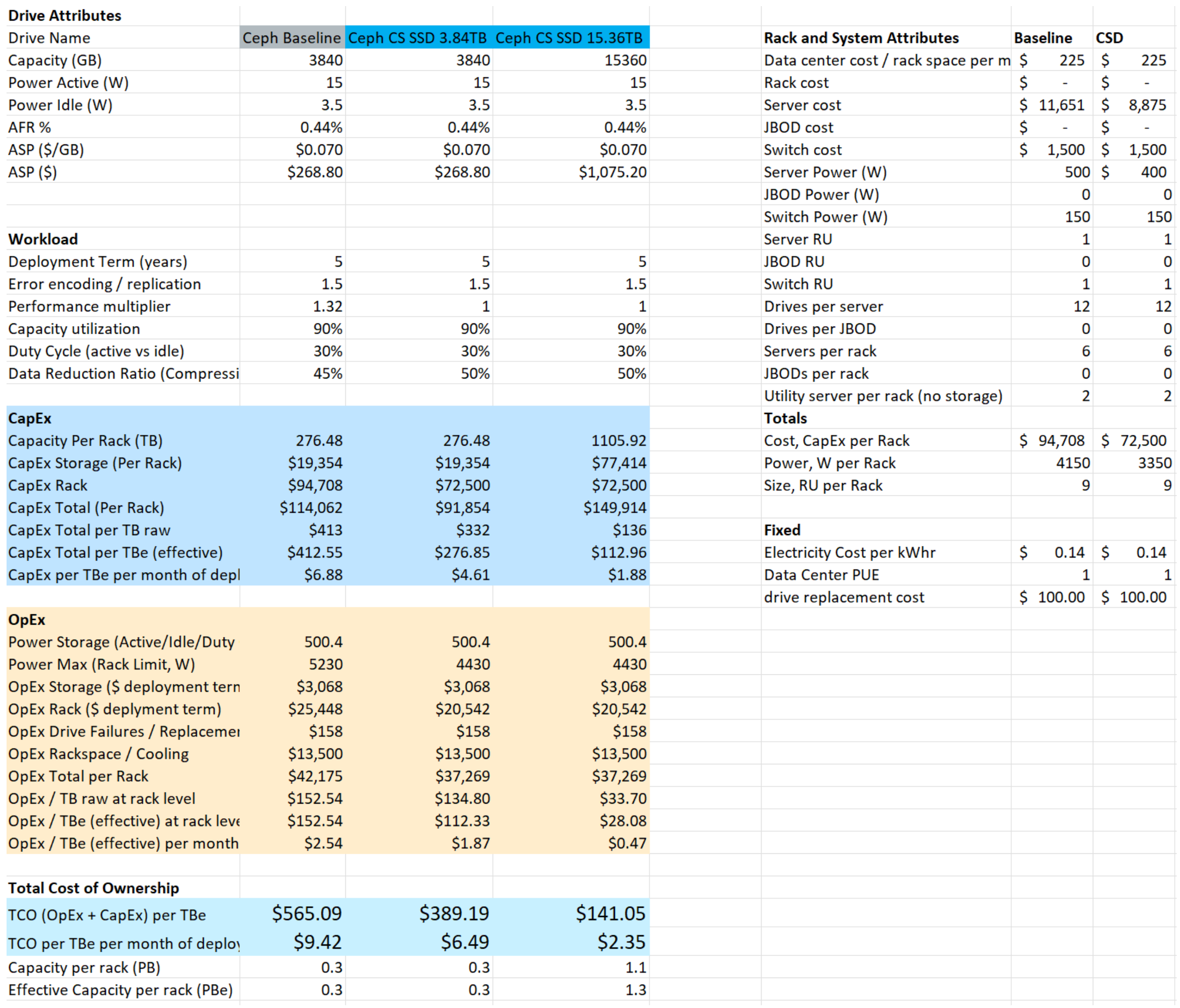
```
bdev_enable_discard: True
mgr_max_pg_num_change: 32768
mon_max_pg_per_osd:
mon: 32768
osd: 32768
mon_osd_max_creating_pgs
mon: 32768
osd: 32768
mon_osd_max_initial_pgs
osd: 32768
```

Fio Configuration

```
[global]
ioengine=rbd
direct=1
bs=128k
iodepth=128
rw=write
runtime=1200
pool=test1
time_based
buffer_compress_percentage=60
group_reporting

[rbd_image_1]
rbdname=rbd_image_1

[rbd_image_2]
rbdname=rbd_image_2

...

[rbd_image_40]
rbdname=rbd_image_40
```

TCO Model Configuration

A link can be found here



Drive Attributes			
Drive Name	Ceph Baseline	Ceph CS SSD 3.84TB	Ceph CS SSD 15.36TB
Capacity (GB)	3840	3840	15360
Power Active (W)	15	15	15
Power Idle (W)	3.5	3.5	3.5
AFR %	0.44%	0.44%	0.44%
ASP ($/GB)	$0.070	$0.070	$0.070
ASP ($)	$268.80	$268.80	$1,075.20
Workload			
Deployment Term (years)	5	5	5
Error encoding / replication	1.5	1.5	1.5
Performance multiplier	1.32	1	1
Capacity utilization	90%	90%	90%
Duty Cycle (active vs idle)	30%	30%	30%
Data Reduction Ratio (Compressi	45%	50%	50%
CapEx			
Capacity Per Rack (TB)	276.48	276.48	1105.92
CapEx Storage (Per Rack)	$19,354	$19,354	$77,414
CapEx Rack	$94,708	$72,500	$72,500
CapEx Total (Per Rack)	$114,062	$91,854	$149,914
CapEx Total per TB raw	$413	$332	$136
CapEx Total per TBe (effective)	$412.55	$276.85	$112.96
CapEx per TBe per month of depl	$6.88	$4.61	$1.88
OpEx			
Power Storage (Active/Idle/Duty	500.4	500.4	500.4
Power Max (Rack Limit, W)	5230	4430	4430
OpEx Storage ($ deployment tern	$3,068	$3,068	$3,068
OpEx Rack ($ deplyment term)	$25,448	$20,542	$20,542
OpEx Drive Failures / Replacemer	$158	$158	$158
OpEx Rackspace / Cooling	$13,500	$13,500	$13,500
OpEx Total per Rack	$42,175	$37,269	$37,269
OpEx / TB raw at rack level	$152.54	$134.80	$33.70
OpEx / TBe (effective) at rack leve	$152.54	$112.33	$28.08
OpEx / TBe (effective) per month	$2.54	$1.87	$0.47
Total Cost of Ownership			
TCO (OpEx + CapEx) per TBe	$565.09	$389.19	$141.05
TCO per TBe per month of deploy	$9.42	$6.49	$2.35
Capacity per rack (PB)	0.3	0.3	1.1
Effective Capacity per rack (PBe)	0.3	0.3	1.3

Rack and System Attributes	Baseline	CSD
Data center cost / rack space per m	$ 225	$ 225
Rack cost	$ -	$ -
Server cost	$ 11,651	$ 8,875
JBOD cost	$ -	$ -
Switch cost	$ 1,500	$ 1,500
Server Power (W)	500	$ 400
JBOD Power (W)	0	0
Switch Power (W)	150	150
Server RU	1	1
JBOD RU	0	0
Switch RU	1	1
Drives per server	12	12
Drives per JBOD	0	0
Servers per rack	6	6
JBODs per rack	0	0
Utility server per rack (no storage)	2	2
Totals		
Cost, CapEx per Rack	$ 94,708	$ 72,500
Power, W per Rack	4150	3350
Size, RU per Rack	9	9
Fixed		
Electricity Cost per kWhr	$ 0.14	$ 0.14
Data Center PUE	1	1
drive replacement cost	$ 100.00	$ 100.00

SNIA. CMSI | COMPUTE, MEMORY, AND STORAGE

SNIA
5201 Great America Parkway, Suite 320, Santa Clara, CA, 95054
Phone:719-694-1380 • Fax: 719-694-1385 • www.snia.org

SNIA