StorPool
DISTRIBUTED STORAGE

The 2024 Block Data Storage Buyer's Guide

A Pragmatic Process to Selecting Primary Shared Block Data Storage

The 2024 Block Data Storage Buyer's Guide

A Pragmatic Process to Selecting Primary Shared Block Data Storage

This guide was created for CEOs, CTOs, CFOs, and business leaders who are looking for a comprehensive understanding of a better approach to data storage for their business. An approach that will help to gain control of overall operational costs, ensure the right data storage for business needs, and position infrastructure to handle any amount of data growth today and in the future.

Additionally, IT Team Leaders, storage architects, and DevOps leaders will gain a comprehensive understanding of evaluating the right data storage based on need-to-haves and want-to-haves to not only ensure a modern block storage system but a system with the right automation to give them the confidence in their daily routines that the storage is always working and delivering data when and where it is needed allowing IT teams to have the time to focus on the next innovation and the real business at-hand.

About the Author

Marc Staimer, as President and CDS of Dragon Slayer Consulting in Beaverton OR since 1998, is well renowned for his in-depth and keen understanding of IT technology problems, especially with storage, networking, applications, cloud services, data protection, and virtualization. Marc has published countless technology articles and tips from the IT user perspective for internationally renowned online publications including Wikibon, TechTarget's Searchxxx.com websites, and Network Computing.

Introduction

There are many storage buyer's guides readily available on the internet. So why publish a new one? An extensive analysis of those storage buyers guides reveals they're not very useful. They tend to be too rudimentary. They assume quite a bit, are too general, and fail to provide down-to-earth steps necessary to determine the best, most reliable, and most cost effective storage for the buying organization.

They're generally organized into three sections. Section 1, which tends to be the most useful, will be discussed shortly. Section 2 often provides a brief list of some of the prominent vendor storage systems including a brief description, a list of features, and a few pros and cons. The information is hardly comprehensive. The data storage systems' descriptions are quite general with no more insight than what's on the vendor's website. It tends to be much too shallow to be of much use.

Section 3 typically changes the focus to features the buyer should be looking for based on the author's opinion. They are not on the problems they solve. That's highly problematic without the proper context. Some are merely feature checklist comparisons. No discussion on what those features solve - feature value. They list just whether they have them or not. System A has 31 features and system B has 32. That makes system B better? Without context it cannot be determined. When the buyer's guide does this, it comes across more as infomercial versus a useful data storage unbiased buyer's tool.

Then there is the first and most useful section. Section 1 is useful in that it counsels what is generally needed for IT preparations in purchasing and utilizing a new data storage system. That is the only similarity between other storage buyer's guides and this one. This demands specific knowledge. Instead of devoting a complete section to this fundamental, task to basic knowledge such as knowing:

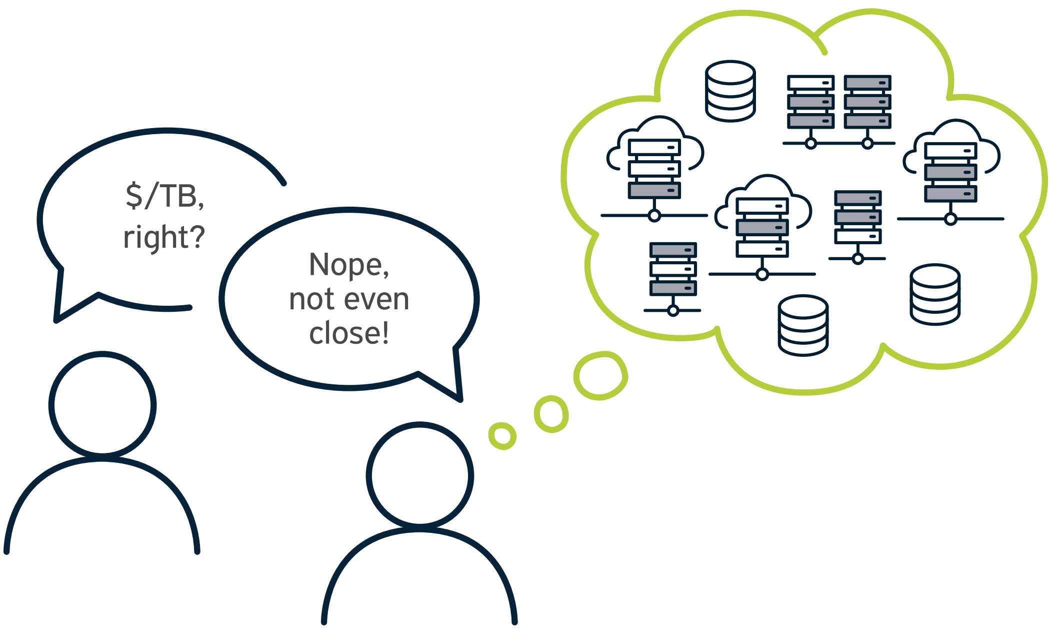
- The capacity requirements for both current data being stored and retained in addition to all projected future data storage for the life of the system.
- The type of workloads that will be using the data storage. They will vary. This essentially breaks down into application response times to the user. The response time is affected by the application server performance and available resources, networking between the user and the application server, networking between the application server and the data storage, in addition the data storage system performance and resources.
- Each of the workload requirements for performance in latency, IOPS, and performance: capacity; and data protection - recovery point objectives (RPO)[1] and recovery time objectives (RTO)[2].
- Government regulations, of which there are many. Some are industry specific like HIPAA or Basel II and III. Others are more general such as GDPR or CCPA, but increasingly common for data privacy, data sovereignty - data kept in a specific country, geographic area, or on-premises.
- Storage security processes for preventing or at least mitigating breaches, data theft, malware infiltration, ransomware attacks, unauthorized access, accidental deletions, software corruptions, and malicious employees.
- Current or planned IT infrastructure on-premises and/or in the public cloud requirements. If the application workloads require NVMe/TCP at 25Gbps Ethernet and the current network is primarily 1Gbps or even 10Gbps, something has to change.
- Current and desired timeframe objectives when provisioning data storage for new applications and devops. i.e., how long do the application owners and devops people have to wait? Too long and they will find their own storage. This is called 'Shadow IT'. Shadow IT is a problem for IT

1 Amount of data that can be lost. RPOs reflect the time between data protection or backup events.
2 How fast the workload needs to be back up and running.

organizations. Support eventually falls to them even when they had no say in the selection of the data storage, no training, and no staff for it. An outage is commonly the event that pulls IT into taking over the support.

- A frank assessment of internal data storage knowledge, skills, and experience of the IT administrators responsible for the new data storage system. The key word here is...frank.
- Timeframes required for the new system to be up and running. As the Covid 19 pandemic made clear, supply chains can be and are disrupted at times. Some vendors are better prepared than others.
- The budget for the new storage system - including all software and hardware; implementation costs; professional service costs; ongoing costs for maintenance, subscription, or Storage-as-a-Service (STaaS); supporting infrastructure costs such as rack units (RU) and the allocated fixed data center overhead assigned per RU, power, cooling, UPS, switch ports, cables, conduit, transceivers, cable management, and their associated maintenance or subscription costs. One other cost that must be budgeted is the tech refresh cost at the data storage system's end-of-life. That includes the data migration cost, professional services cost, full time employee (FTE) cost spent on the tech refresh, and the co-resident cost for both data storage systems while the tech refresh takes place. This cost is often left out of the budget.
The budget decision should not be limited to just a price per terabyte measure. Price per terabyte assumes that all other costs are equal. That is a very inaccurate assumption. Price per terabyte is horribly misleading and leads to poor decisions. It assumes there is only a single performance tier and all media is the same cost. That's a fantasy. Every type of media has different performance characteristics, capacities, and cost. That's led many storage buyers to use 3 different measures per storage tier:

1. TCO (total cost of ownership) over the data storage system's life per TB.
2. TCO per IOPS.
3. TCO per throughput in bytes per second.



Objective: To Provide Buyers a Rational Block Data Storage Buyer's Guide

Making this Data Storage Buyer's Guide more pragmatic compels considerable differences from all the others currently available. It has to be simple and useful.

The 1st difference is the focus on block data storage. It's in the title. Why block? Because block data storage is the principal choice for performance. Most primary data storage is in fact block.

That doesn't mean this buyer's guide ignores file and object data storage. On the contrary, it explains the differences in storage types, use cases, and problems they solve. However, there is much more detail on the block storage problems, workarounds, capabilities data storage buyers should be looking for - depending on their use cases, and how they should evaluate each vendor's data storage offerings.

The 2nd difference is the effort to educate and debunk data storage myths. There are a lot of them.

The 3rd major difference is the emphasis on the 'why'. Why certain capabilities and features? The 'why' are the problem details solved by block data storage. Why workarounds to some block data storage problems fail, are unsustainable, and have ultimately very high costs.

The 4th extremely valuable difference is a useful, simple, and pragmatic process tool that empowers data storage buyers to compare and contrast different vendor block data storage systems. It is designed to compare how well each block data storage system solves the block data storage problems, workarounds, and TCO.

The 5th fundamental difference is the detailing of a simple intuitive process for calculating real TCO for each block data storage system. Too many of these data storage buyers guides emphasize the purchase prices, subscription fee, or cloud storage-as-a-service fees. Those costs are often referred to as price/TB.

Table of Contents

Differences Between Block, File, Object, and Unified Data Storage

Block data storage

Block data storage is the universal underlying storage infrastructure. All data block, file, or object data is ultimately stored in a block format on the storage media. That data is saved to storage media - SSD, HDD, tape, or Optical - in fixed-sized chunks called blocks. A data block is a sequence of bytes or bits. It has a fixed length known as block size. It generally contains a whole record number. Each block has a unique address. That address is the only metadata assigned to each block. Block management is handled by software that controls how blocks are placed, organized, and retrieved correlated to the metadata.

Data is stored on the internal media of a server and workstation or connected to an external block data storage system over a network - Historically called a storage area network (SAN). Note: the more common name today is storage network.

In block data storage systems each individual storage volume acts as an individual hard drive configured by a storage administrator. These systems are connected to application or database servers in multiple ways.

1. Direct attached storage or DAS - i.e. external SAS, FIbre Channel (FC), or iSCSI ports on the storage controller.

2. High performance storage switched networking a.k.a. NVMe_oF:
    - NVMe/RoCE - layer 2
    - NVMe/FC - layer 2
    - NVMe/IBA - layer 2
    - NVMe/TCP - layer 3
    - NVMe/NFS - layer 3

3. Standard storage switch networking a.k.a. SAN:
    - SCSI over FC - layer 2
    - iSCSI over Ethernet - layer 3
    - SCSI over IBA - layer 2

Block data storage is ideal for high-performance, mission-critical, data-intensive, and Enterprise applications needing consistent low latency high I/O performance. Applications such as relational databases, online transaction processing (OLTP), eCommerce, or any application that demands subsecond or real-time response times. It is also ideal for high speed analytics. In essence, when low latency, high IOPS, or very high throughput (measured as bytes per second) are the requirement, block data storage is more often than not the fastest shared data storage.

Block data storage is data agnostic, working well for all structured and unstructured data types, hypervisor virtual drives, and persistent container storage. The strengths of block data storage are performance and flexibility. The weaknesses tend to be technical complexity and cost.

File data storage

File data storage is a hierarchical storage methodology. It can use either SSDs or HDDs. It's also known as file-level or file-based storage. This type of storage writes, organizes, stores data as files with the files organized in folders with the folders organized in a hierarchy of directories and subdirectories. Files are located via a path from directory to subdirectory to file. This organization occurs on the internal media of a server and workstation or a network-attached shared file storage (NAS) system. That connection is via layer 3 networks - TCP/IP or NVMe/NAS over Ethernet or InfiniBand.

File data storage is very well suited for unstructured file data, PACS DICOM health imaging data, backup data, cool data, and even cold data. It is ideal for unstructured data. However, it works adequately for structured data including many database types, hypervisor virtual drives, and persistent container storage. Its biggest strength is its simplicity.

File data storage historically had trouble scaling to very large file counts. That's changed in the past several years. File data storage can now scale to many billions even trillions of files through clever metadata management.

The downside to file data storage is its I/O performance is noticeably lower than block data storage. It's hampered by the additional file storage and management software layer latencies. That performance limitation reveals itself in reduced application response time, concurrent IO, and total throughput.

The exception to total throughput being lower is the parallel file system (PFS) data storage. PFS aggregates or bonds multiple file data storage servers simultaneously enabling very high throughput. It's complicated and is primarily utilized in high performance computing and large unstructured data ingestion.

Object data storage

Object data storage organizes, stores, and manages data as discrete units or objects. Unlike files, they're stored in a single repository. There is no need for nesting of files inside a folder that may be inside other folders. In other words, there is no hierarchical structure. All data is stored into a flat address space known as a storage pool or bucket. The blocks that make up an object are kept together while adding all of its associated metadata. One object data storage advantage is the ability to add extensive and unique metadata to the objects. Users can define their own metadata. File storage generally cannot. (There are some limited exceptions.) Every object - file, photo, video, etc. - has a unique identifier. That metadata is an essential key to its value enabling extensive data analytics of the object data.

Similar to file data storage, object data storage is connected via layer 3 networks - TCP/IP over Ethernet. Object data storage can use both SSDs and HDDs. But primarily HDDs because object storage is not known for being much of a performance play but rather a lower cost data storage play. SSDs only nominally affect the performance issues of object storage. Object data storage I/O latency is notably higher than file and especially block data storage.

Object data storage is primarily utilized as secondary or tertiary storage for cool and cold data. It is mostly used for unstructured data found in data lakes - large amounts of cool data used in analytics and machine learning, public clouds, backups, and archives. It competes with file data storage while making inroads in the DICOM PACS imaging healthcare markets as well as the media and entertainment markets.

Unified data storage

Unified data storage is the combination of all three storage types or two of the three. A combination of block and file data storage or file and object data storage are the most common unified data storage types.

There is a problem with most proprietary and open-source unified data storage systems. They are predominantly developed for one type of data storage. The other types are gatewayed. Gateways add latency and reduce application performance. One proprietary file data storage system converts blocks to files. Then the files are written ultimately as blocks. Remember, both file and object storage ultimately writes data to the media as blocks. Converting blocks to files - in order to leverage file storage services - adds another layer of latency. Another open-source object data storage converts both files and blocks to objects before eventually writing the data to the media as blocks. Again, two layers of latency.

Unified data storage works best when the majority of the data is written to its primary data storage type. The other data storage types are mainly for convenience, not performance purposes.

Differences Between Software-Defined Storage and Data Storage Systems

Data Storage Systems

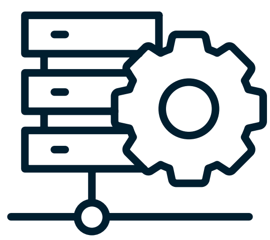
Buying data storage systems or renting them via storage-as-a-service (STaaS) means getting a complete integrated and tested system. Everything is integrated, burned in, and optimally aligned. Everything works. Support is a single vendor meaning one throat to choke when there's a problem. And there are always problems.

The downsides are much higher costs; vendor lock-in - can't buy software, parts, or maintenance from anyone else; significant lag time in months or years behind server implementation of CPU, networking, and storage media innovations - frequently skipping generations. The innovation lag is because the storage vendor needs to get a ROI on each released model and that typically requires a minimum of 3 years.

Traditional
Data Storage Systems

Storage innovation
available every 3-5 years

Storage Software Converts
Servers into Storage Systems



Storage innovation
available every year

Software-Defined Storage (SDS)

Software-defined storage (SDS) extracts the core data storage operational software from the physical hardware. This enables the software to run on bare metal common-off-the-shelf (COTS) x86 servers, a.k.a. white box, or brand name x86 servers, virtual machines (VM), containers, or HCI. Gartner describes SDS as "storage controller software abstracted from the underlying hardware, so it can run on any hardware, any hypervisor, or on any cloud."

This offers several advantages over the data storage system including:

- No hardware vendor lock-in.
- Hardware can be upgraded at any time.
- Current server contracts and discounts can be leveraged.
- Same media is at much lower server, not data storage system prices.
- Better control.
- Much more flexibility.

Downsides of SDS are hardware troubleshooting unless the SDS vendor takes ownership of the problem regardless of whether the problem is in the software or the hardware.

Setting the Record Straight on Data Storage Media

There are four generally available data storage media as of 2023. They are solid state drives (SSD), hard disk drives (HDD), linear tape-open (LTO) tape, and optical.

Neither LTO tape nor optical are much used beyond archiving today. Optical is slow with low capacities. LTO tape usually resides in tape libraries and offline making access slow.

With the demise of Intel Optane, there are no commercially available non-volatile memory (NVM) data storage media beyond Flash NAND or Flash NOR - NAND is by far the dominant Flash media. There are several NVM technologies in development; however, none have made it to volume production yet. That leaves SSDs and HDDs as the prevalent interactive storage medium today.

SSDs

Flash NAND SSDs come in several shapes, form factors, bits-per-cell, capacities, and write life referred to as drive writes per day (DWPD). DWPD refers to the number of complete writes of the total capacity of the drive per day guaranteed by the manufacturer. NAND cells are arranged in program erase blocks. Once data is written to one of these blocks no additional data can be written to that block. This is true even when there is unwritten capacity on that block. To write any additional data to the program erase block requires a layer of material to be erased. It is a destructive process. That is why NAND flash has a limited number of writes.

Manufacturers deal with this by concatenating writes to consume entire program erase blocks and overprovision the SSDs with additional capacity to account for cells that fail or reach the end of their write life. That overprovisioning is not reflected in the SSDs usable capacity.

The most common Flash NAND SSD is in the standard 2.5" drive form factor. The other common form factor is the 3.5" standard. But because Flash NAND are chips, they're not limited to standard drive form factors, which are artifacts from HDDs. There are M.2, pencil, EDSFF (E1.L, etc.) and custom form factors. A few data storage system vendors use their own custom Flash NAND SSD form factors. They assert it lowers cost and provides better write life. Both claims are highly debatable.

Flash NAND SSD capacity, performance, error correction, and write life are all affected by the number of bits in each cell. Each additional bit per cell reduces its write-life - a.k.a. program erase cycles (p/e), reduced write/read performance, increased power required to write or read, and increased errors per write/read requiring more error correcting code. There are 4 currently commercially available Flash NAND SSD types:

1. Single-level-cell (SLC) 1 bit per cell

SLC is by far the fastest and has recently been reclassified as storage class memory (SCM) by several data storage vendors. SLC has the highest write life at around 100,000 writes per cell, lowest need for error correcting code, and highest price per terabyte (TB).

2. Multi-level-cell (MLC) 2 bits per cell

MLC was the first attempt to reduce Flash NAND SSD cost. Write life decreased to ~ 10,000 to 30,000 writes per cell. It has largely fallen out of favor because of more bits per cell, lower cost alternatives.

3. Triple-level-cell (TLC) 3 bits per cell

TLC appears to have achieved the correct balance between performance, write life, and cost even with significantly more overprovisioning than SLC or MLC. Write life per cell is ~ 1,000 to 3,000 writes per cell. An important Flash NAND manufacturing innovation has been 3D layering that's become common for TLC SSDs. This has enabled much larger capacities per TLC SSD - as high as 100TB in a 3.5" form factor - while lowering the price per TB.

4. Quad-level-cell (QLC) 4 bits per cell

QLC is primarily used as a competitor for nearline HDDs. It has a limited write life of ~ 100 to 300 writes per cell meaning the number of DWPD are severely limited. QLC also takes advantage of 3D layering to deliver higher capacities - as high as 64TB in a 3.5" form factor and one coming out at 128TB in a 2.5" form factor - at lower price per TB than even TLC. Performance is also noticeably lower than TLC.

Several storage vendors pushing all-flash arrays claim their QLC drives are cost equivalent to nearline HDDs. Even when considering HDD greater consumption of power, cooling, and rack units (RU) for the same capacities, those claims should be viewed with significant skepticism. Make them prove it. Also make them spell out specifically what they're comparing. Several recent studies put QLC SSDs at 5 to 7 times more expensive than high capacity HDDs. That difference is forecasted to decline to be only 4 to 5 times more expensive by decade's end.

Every realistic honest comparison shows Nearline HDDs are still considerably less costly. This is critically important since the amount of organization cool or cold data is typically ≥ 9x that of hot data. QLC will definitely outperform HDDs in transactional read/write performance. But sequential reads/writes is a different story where the performance differences are negligible.

HDDs

HDDs have been whittled down to the 3.5" form factor Nearline drives. SSDs killed off 2.5" form factor 15,000 RPM, and for the most part, 10,000 RPM HDDs. Nearline HDDs are 7,500 RPMs or less. These are known as high capacity or fat drives. Mostly used for cool and cold data - backup and archive, current capacities top out at 25TB in a 3.5" form factor with a road map to 100TB in the next few years.

Will QLC SSDs or possibly future PLC (penta-level-cell or 5 bits per cell) SSDs replace HDDs? Not now, but possibly sometime down the road. It's a bit unlikely in the next few years, contrary to some vendor assertions.

The Fundamental Application Problem Uniquely Solved by Block Data Storage

Application response time. Data storage performance metrics do not refer to application response times. It should. It only talks about data storage performance. Performance metrics are measured as latency - i.e. delay, IOPS - I/Os per second, and throughput - bytes per second or Bps. These performance metrics ultimately affect application response times.

Applications that demand lower latency and high IOPS for faster response times will generally connect to block data storage. No other data storage type matches block data storage in latency and I/O performance. Throughput is debatable specifically when comparing parallel file data storage that binds multiple ports and data storage controllers/servers. But there is no question that block storage is preferred when low latency, high IOPS, and greater transactions per second are required. In other words, structured data applications. This is why the cloud data storage for most transactional applications is block data storage.

Some may question why they may need data storage with high performance, low latency, and faster I/O. They might think their current data storage performance is good enough. They're not thinking about application response time. More specifically, sub-second application response times. They don't realize or understand that sub-second response times lead to much greater user productivity, higher morale, faster time-to-market, faster time-to-actionable-insights, faster time-to-revenues and unique revenues, lower headcount, and higher profits. IBM research demonstrated this unequivocally in 1982 with their Red Book called, The Economic Value of Rapid Response Time.

> *"When an application and its users interact at a pace that ensures that neither has to wait on the other, productivity soars, the cost of the work done on the application's computer infrastructure tumbles, users get more satisfaction from their work, and their quality improves."*

The research further revealed that as application response times reached ≤ 400ms, which is four tenths of a second, productivity soared, and users became addicted to that response time. This is called the Doherty Threshold. It's cleverly explained in a clip of an episode of the television series "Halt and Catch Fire," on YouTube.

More detail about this IBM research can be found in Appendix A.

The most important takeaway from that research is the incredible importance of low latency, high performance I/O block data storage. It leads to lower costs, faster-time-to-market, faster-time-to-actionable-insights, faster-time-to-unique-revenues, and-faster-time-to-substantial-profits.

Failure to utilize such storage can and often does result in reduced application response time, low productivity, late-to-market products and services, reduced revenues, and increased costs. That translates into a competitive disadvantage for both users and hosting providers.

Keep in mind there are several shortcomings and problems with many block data storage systems and software. Working around or solving those problems is crucial to leveraging the power of block data storage.

Block Data Storage Shortcomings, Workarounds, Consequences, and How to Better Solve

The biggest block data storage problems center decidedly around performance restrictions, non-scalable performance, capacity scalability boundaries, compromised data protection, extensive software inefficiencies, unnecessary complexity/lack of automation, and excessive cost.

Performance restrictions

Block data storage system and software performance has a history of declining when capacity utilization exceeds 50%. It falls off a cliff at 80% utilization. It's why most vendors recommend never to exceed 80% capacity utilization regardless of the amount of capacity installed. Avoiding that threshold frequently requires the addition of more capacity. What it really means is that at least 20% of your capacity is wasted and can never be consumed at all times. Please note, that the 80% capacity utilization threshold is typically the same for file and object storage.

Workaround: **buy extra capacity that you won't actually use**

This workaround alleviates capacity utilization performance degradation, but it increases the amount of unusable capacity. There is a hard limit to this workaround with every storage system or SDS. It ultimately is a costly and unsustainable workaround.

Workaround: **buy block data storage systems that scale beyond 80% of your perceived requirements**

The problem with this workaround is that it adds significant cost without addressing the performance declination above 50% utilization. It only pushes out the 80% cliff.

Better Solution: **buy block data storage systems that utilizes ≥ 90% of capacity without performance degradation**

There are a few block data storage systems or software that can do this today. They all should.

Non-scalable performance

Most block storage systems and software are limited to two active-active, or active-passive data storage controllers or servers. In either case, each controller/server has access only to the data it specifically stored. It is when one of the controllers/servers fails that it can access the other's stored data. There are some exceptions to this norm.

What it means is the active-active block data storage aggregate performance is somewhat restricted to that of a single controller/server or at best two. That might be adequate for your current and perceived growth rates. If it's not, then how do you solve it?

Workaround: buy more block data storage systems

There are several problems with buying more block data storage systems. It's not just the cost, although cost is a major issue. A bigger issue is the management. Each new data storage system adds more than 100% more management tasks. These are duplicate tasks that humans are not very good at. And then there are the additional tasks that grow increasingly complicated as more and more systems are added to address performance.

Workaround: Go All-flash or faster flash

If the data storage media is the performance bottleneck, that can work. NVMe flash SSDs have lower latency, higher IOPS, and higher throughput than SAS or SATA SSDs. However, flash SSD performance is rarely the performance problem root cause. Need proof? Add up the aggregate I/Os per second and throughput performance of all of the flash drives in an all-flash array. It's likely greater than the rated performance of the all-flash array. Although NVMe will bypass SAS or SATA controllers reducing overall latencies, the bigger most consistent performance bottleneck is the data storage controller/server.

The reason behind the data storage controller/server performance bottleneck is the de facto standardization on the x86 CPU. The x86 CPU became the de facto standard because of Moore's Law of doubling the transistors and performance every two years. It enabled storage software to be inefficient because it was covered up by the exponential growth in x86 performance every couple of years. But all good things come to an end and Moore's law has been slowing substantially. Now, instead of performance doubling, the trend is to double or at least increase the number of cores with only marginal improvements in the performance of each core. And the timeframe for each new CPU has increased beyond 2 years. All of this leads to data storage controllers/servers that are not improving all that much year over year.

The non-scalable data storage performance culprit is more likely the data storage controller/server, not the media, nor whether or not the data storage array is all-flash. Then take into consideration that data storage controllers are generally limited to 2 sockets (CPUs), most block data storage systems or software are limited to 2 data storage controllers/servers, and it does not take a rocket scientist to identify the bottleneck.

Better Solution: buy scale-out block data storage

Scale-out block data storage solves the limited number of storage controllers/servers and eliminates that performance bottleneck. Scale-out can be shared nothing or shared everything. Shared nothing scale-out block data storage tends to have better performance with lower latency and higher IOPS than shared everything. Shared everything will have lower scalability limits than shared nothing. Both solve the limited controller performance bottleneck problem.

Compromised data protection

Data storage data protection is table stakes today. But what that encompasses varies considerably among data storage systems and software.

Snapshots

A good example of compromised data protection is snapshots. Snapshots have been touted as an excellent way to protect data on a data storage system. But the number of snapshots supported per volume has become inadequate in the modern data center. That snapshot number is critical in determining recovery point objectives (RPO) a.k.a. the amount of data that can be lost. RPOs are the time between snapshot events. When snapshots are taken once a day then the RPO is 24 hours. That's how much data will be lost in an outage requiring a recovery. When it's four times a day, the RPO is 6 hours. More snapshots per volume enables smaller RPOs. Fewer snapshots mean larger RPOs. Too many block data storage systems or software have limited snapshots per volume. This becomes critical when snapshots need to be retained for very long time periods.

A common snapshot limitation is approximately 256 per volume. Where that may seem like a lot, a little math shows why it's not. If the snapshot retention policy is a year, 256 is only going to allow one snapshot approximately every 34 hours. That's less than a snapshot a day with a very high RPO. Even when the retention is just 30 days, it works out to be at best one snapshot every 2 hours, 48 mins, and 45 secs. Not exactly a small RPO. Especially if you need RPOs in single digit minutes.

Why the limitation? It comes down to data storage snapshot inefficiencies. When snapshot software is inefficient, and most are, each snapshot consumes a large amount of data storage controller/server resources. Snapshots are extremely fast. However, enough of them can cause a noticeable significant decrease in data storage I/O performance. That's the foremost reason snapshots are limited.

Workaround: **Use a 3rd party data protection product and services**

There are several 3rd party data protection products and services that provide small RPOs. However, they are essentially fixing a data storage system feature deficit for an oversized cost.

Better Solution: **block data storage that provides lots of snapshots**

Block data storage system that delivers large or unlimited numbers of snapshots enabling smaller RPOs. Doing so effectively will not negatively impact the I/O performance.

Ransomware & Malware

Ransomware has become a considerable threat to all organizations. They can and will delete data storage snapshots and backups. This has made snapshot immutability important. Immutability prevents the data from being corrupted, changed, or deleted. Preventing the malicious ransomware from stealing admin credentials that could override that immutability calls for two factor authentication (2FA). Few block data storage systems provide both today.

Malware and ransomware can both copy and steal your most sensitive and mission-critical data. Data breaches will hurt your organization in litigation, painful regulatory fines, reputation, lost customers, revenues, and massive repair costs. Data storage can mitigate these events with 2 factor authentication (2FA) and internal data encryption. 2FA prevents the ransomware or malware from copying out your data without having compromised both devices. The internal encryption prevents physical access to the data stored on the media or unauthorized reading of the data. This is especially important when a drive fails and is retired.

One more data storage ransomware mitigation feature is anomaly detection. This capability essentially detects an anomalously high change rate in the data, which occurs when ransomware is encrypting the data. The data storage system can provide an alert that this is going on. The storage admin can then take steps to stop the encryption, block the offending ransomware, and initiate a recovery from the latest immutable snapshot.

Good Solution: **Use a 3rd party data protection product or services**

There are several 3rd party data protection products or services that can provide small RPOs, immutable backups, 2FA, and encryption. Many database applications will also store their data encrypted. The best 3rd party data protection applications even proactively scan the backed up data to eliminate infected, but not yet detonated ransomware and malware.

Good Solution: **block data storage with built-in snapshot immutability, 2FA, and internal encryption**

Block data storage snapshot immutability can prevent ransomware from corrupting, changing, or deleting snapshots. 2FA prevents ransomware or other malware from accessing, stealing, or breaching your data. Internal data storage encryption prevents physical access to the data. This last defense prevents a disgruntled or malicious employee from taking the media and the data on it. It also provides security of the data when a drive fails and has to be disposed of. And it prevents unauthorized software applications from reading the data.

Failed drive rebuilds decimate performance

RAID has been the block data storage array's failed drive protection since the early 2000s. RAID was effective when drive capacity was measured in megabytes or gigabytes. Now that drive capacity ranges up to 100 terabytes, RAID has become a problem.

Unfortunately, RAID decimates I/O performance during a drive rebuild. A single failed drive can reduce I/O performance by as much as half. Two concurrent drive failures being rebuilt in RAID 6, will reduce data storage system performance by as much as 80%. Larger capacity drives (15TB, 30TB, 100TB, 128TB SSDs and 18TB, 20TB, 22TB HDDs) take much more time to rebuild. Time that often stretches into weeks.

General Just Ok workaround: **Run the drive rebuilds in background**

Running a drive rebuild in the background will minimally reduce I/O performance. Much less so when there are two concurrent failed drives being rebuilt. The problem with running RAID rebuilds in the background is that it doubles, triples, even quadruples rebuild time. That exposes data being rebuilt to greater risk of data loss should another drive in the RAID 5 group, or two more drives in the RAID 6 group fail. That is statistically more likely to happen than most IT professionals recognize.

General Just Ok workaround: **Recover any RAID group failures with Snapshots**

Snapshots are generally an ok workaround. Most IT organizations would prefer not to experience a preventable outage because of the snapshot RPO issue. There will be lost data. And an outage means there is some level of downtime. Downtime is very costly in productivity, lost revenues, customers, stock value, and reputation.

Much Better Solution: **block data storage with fast rebuilds, low resource intensity, erasure coding**

Erasure coding has been touted as the next generation of RAID. It breaks blocks of data into chunks and places them on different drives behind different data storage controllers/servers. The number of concurrent tolerated drive failures is often configurable and flexible. It also protects the data from one or more data storage controller/server failures. Erasure coding does not rebuild the drive, it rebuilds the data on available capacity on multiple drives. That shortens rebuild times considerably. Data rebuilds are accelerated as more data storage controllers/servers are utilized for the rebuilds.

The problem with erasure coding is that it is resource intensive. It needs a lot of data storage controller/server resources to write, read, and rebuild the data that was on a failed drive or data storage controller/server. That in turn greatly reduced I/O and throughput performance. Thus, relegating it to secondary storage for cool and cold data such as scale-out file or object storage. But not block storage.

That is no longer the situation. A few block data storage suppliers have solved the erasure coding resource requirements, making it much more efficient. In this new generation of erasure coding there is little to no impact on I/O and throughput performance. Additionally, multiple concurrent drive rebuilds occur without reducing I/O and throughput performance or requiring downtime. It's a very good idea to pick a data storage system that provides highly efficient erasure coding.

Time consuming and costly block data storage tech refresh

Tech refresh is a must for all data storage. Whether it be software-defined storage (SDS) or a complete data storage system, the hardware and drives must be refreshed to take advantage of ongoing technology improvements. Improvements in interconnect, NICs, switches, faster and/or denser drives, latest PCIe generation, CPUs, memory, controllers, etc. Data storage technologies are constantly innovating. Leveraging that innovation requires some level of tech refresh, a.k.a. replacing the hardware. Doing so is non-trivial.

Block data storage tech refresh is the bane of every data center. It takes extensive effort, skill, expertise, time, and money. It's complicated and commonly requires at least one outage - downtime. Ongoing research by DSC found the average time to complete a block data storage system tech refresh is approximately 9 months. That's 9 months of duplicate data storage systems on the data center floor being powered, cooled, managed, maintained, and paid for concurrently.

Another Costly Workaround: Buy data storage system that auto-replaces controllers

There are several data storage vendors that offer this type of package on their systems. It provides assurance that their active-active controllers will be replaced every 3 to 5 years as they come out with new releases. There are several flaws with this workaround.

The minimum timeframe for the controller replacement is 3 years. A lot of innovation occurs in those 3 years. The cost of this additional insurance policy is very high. The math shows the total cost of replacing those data storage controllers is 3x more than just buying new controllers. However, doing the latter may incur additional tech refresh time and costs. Additionally, the higher cost of tech refresh is replacing the drives. The drives are typically warrantied for 3-5 years. They will eventually wear out and fail. Replacing the data storage controllers and not the media is very risky.

Always remember the first rule of data storage is to "do no harm to the data". The second rule, remember the first rule.

Just an Ok Workaround: Buy and use hyperconverged infrastructure (HCI)

HCI utilizes its own SDS as its data storage. The upside of this approach is that HCI SDS allows new server nodes to be added to the cluster without a significant effort in tech refresh.

The downsides include an inability of the data storage to be shared outside of the cluster. It generally can't be shared by outside application servers or other HCI clusters. Another downside is the limitation on data protection. For most HCI SDS, snapshots are limited, RAID is the standard, and protection against node failures requires triple copy mirroring at a minimum.

Better Solution: shared block data storage with simple tech refresh

This would allow new data storage controller or server nodes to be added without an outage. Active-active systems would replace the first controller as if it was a failover event. After completion the next controller would be replaced the same way. Drives could be replaced as if they failed and were rebuilt, or if the drawers are not full, put in the new drives, put them in the same volume and RAID group, copy the data from the old drives, remove the old drives from the volume and RAID group.

Scale-out shared block data storage is simpler. Add a new controller with internal drives to the cluster; replicate the configuration from the node to be retired including volumes, data protection, permissions; retire the old node from the cluster. Adding just new drives is also simple. Place the drives in empty drive slots anywhere in the scale-out cluster, configure them into the desired volumes, data protection, permissions, etc. Done.

Extensive software inefficiencies

Data storage controllers/servers have fixed resource constraints. They have limited CPU cycles, memory, and internal bandwidth. And as previously noted, the vast majority of block data storage controllers/servers are active-active or active-passive with each one generally dual socket (2 CPUs).

The problem occurs because data storage software is inefficient. It specifically took advantage of Moore's law, which as previously discussed has been rapidly slowing down. It was assumed that any software inefficiencies would be taken care of by ongoing CPU releases that doubled in performance each time. But those software inefficiencies are no longer covered up by new CPUs. The only way to fix this is to redevelop the data storage software efficiently. Few software developers have any desire to redevelop their code, which is what they have to do to make it efficient. This is one of the reasons you only occasionally see erasure coding in block data storage software.

A Temporary Workaround: Buy and implement more of the same block data storage systems

The most common workaround is to buy and implement more of the same data storage systems. It's NOT a good idea and doesn't work out well. It's based on the misguided perception that adding more of the same types of data storage systems will eliminate additional training requirements while adding nominally more management tasks. That perception is incorrect. Not only do all of the management tasks have to be duplicated, but it also adds plenty of complications. Complications like setting up duplicate volumes, load balancing, and data replication between systems to protect crucial data in the event of a data storage system failure. That complexity and its cost increase exponentially as more and more systems are added. This workaround is ultimately unsustainable.

Better Solution: efficient block data storage software

Much more efficient block data storage software solves a lot of problems. It solves performance problems, data protection problems, and complicated workarounds.

What's better is to marry that efficient block data storage software with non-disruptive scale-out. That combination gets more performance out of less hardware while enabling flexible scaling.

Unnecessary complexity/lack of automation

Far too many block data storage systems require that the administration be done by experts. Storage admins with knowledge, skill, and experience. That philosophical paradigm is problematic because of baby boomer retirements and fewer new hires that can replace those capabilities. The learning curve is steep, and the gap is increasingly problematic.

A Very Costly Workaround: put all your application workloads and data storage in a public cloud

Moving to the public cloud means someone else has to manage the data storage infrastructure. No expertise needed. Lower cost. At least that's what many IT pros believe before they move to the public cloud.

Although they don't manage the data storage, they need to know its strengths and limitations. Take AWS, the largest worldwide public services cloud with 3 provisioned-IOPS flash SSD block volume types:

AWS Provisioned IOPS SSD Volumes – *Multi-EBS Workloads*			
AWS Service	**io2 Block Express ‡**	**io2**	**io1**
Durability	99.999%		99.8-99.9%
Failure/yr.	0.001%		0.1-0.2%
Latency	≤ 1ms	> 1 ms	
Max IOPS	256,000	64,000	
Max Throughput	4 GiB/s	1 GiB/s	

Any application workload requiring more than 256,000 IOPS is completely out of luck. A transactional database would have to be sharded and split over multiple instances with separate io2 volumes to solve. Besides being extremely labor-intensive and time consuming, it's also quite expensive. There are additional database instances for each shard, added hardware infrastructure to run each database shard, and the additional provisioned-IOPS SSD volumes. It adds up quickly and provides quite the bill shock.

A big myth about public cloud data storage is that it's cheaper than on-premises storage. That myth has been persistent and still not true. A thorough DSC analysis revealed that the cloud data storage is based on S3 object storage, not block or file storage. It's not true for object storage either. The analysis was conducted on three different occasions over 4 years compared data storage like-to-like — block-to-block, file-to-file, and object-to-object. The results showed that if you have your own data center or are in a co-lo, the cost of cloud data storage is less costly only for the first 18 months. That's based on street pricing from multiple storage vendors and public clouds. After that, public cloud storage is much more costly. When comparing against storage-as-a-service (STaaS) programs for on-premises data storage, where the data storage is provided in a cloud-like pricing model, the public cloud data storage is more costly from day 1.

An Unsustainable Workaround: Scripts

Scripts are a very common fallback for any system or application that lacks automation. Scripts use the tools and API of the data storage system to provide some of the automation they may need. There are many problems with scripts.

Scripts are rarely documented on paper or within the code. Then there's the lack of both initial and ongoing quality assurance (QA). Scripts are rarely updated when the data storage software is patched or updated. The scripts are mostly updated when they break in production. Once again unlikely to be documented or go through QA testing. The script user interface is usually not made to be intuitive to anyone other than the author.

Then there's the issue of what happens when the person who wrote the script leaves the position or leaves the organization. If it's not documented, and it likely is not, the new data storage administrator may have no idea how the script works or what it does. This leads to the script being completely rewritten, which takes considerable effort and time. Once again, the documentation, QA, patching, testing, etc. tends to be the same.

Better Solution: High levels of automation or STaaS

Extensive automation puts the expertise into the block data storage software and not the administrator. It enables the non-expert and expert alike to get the most optimized system with minimal effort. It greatly simplifies implementations, operations, management, scaling, and tech refresh. The greater the automation the easier and more intuitive it becomes.

STaaS means someone or something else - like automation or AI - is handling and managing everything. From implementation, operations, managing, scaling, and tech refresh. The responsibility for the data storage infrastructure is in the hands of the service provider.

Excessive cost

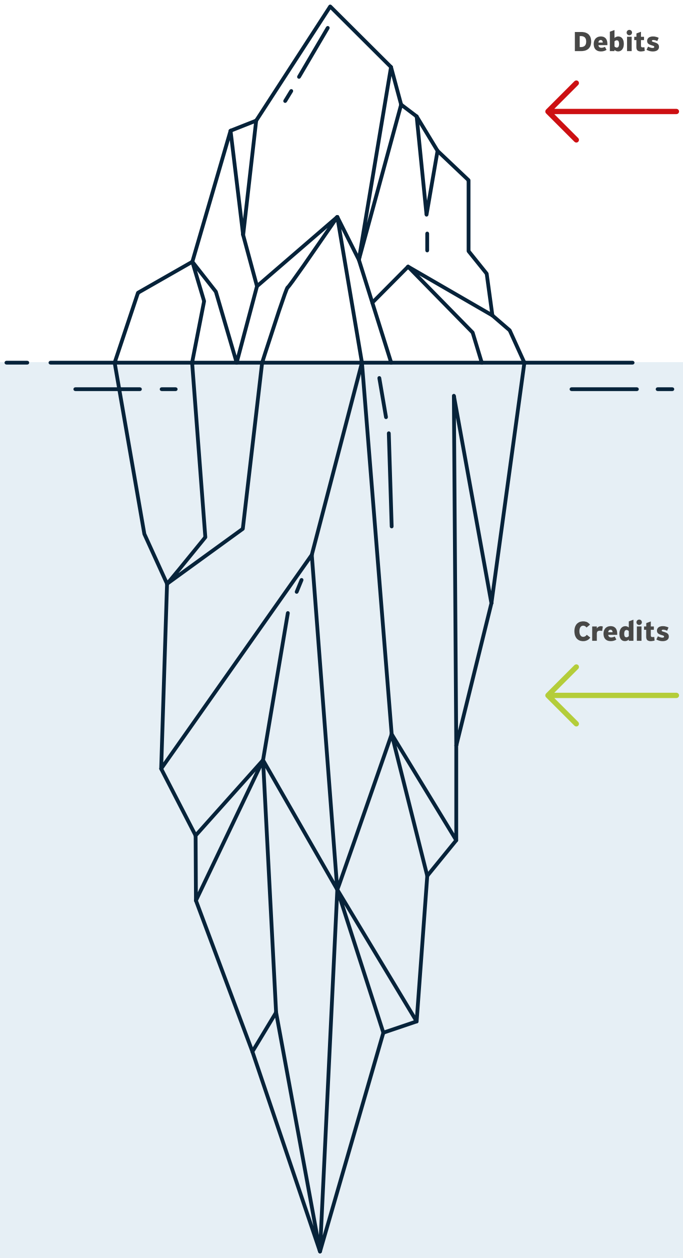
Cost is one of the most misunderstood aspects of block data storage. Too many buyers focus on the single cost aspect of the net purchase price per TB. Focusing on net purchase price per TB is based on too many false assumptions such as:

- **Vendor data storage system discounts are a good deal**
    - Not really. The MSRP on Enterprise and mid-tier block data storage systems are jacked up very high to give the perception that they're giving you a great deal with high discounts. To put that in perspective you need to understand how the system game is played.
    - Enterprise class hardware has an average of 75% discounts in the market. Whereas mid-tier class hardware averages only 60%. Those are both very large discounts"¦or not. Enterprise and mid-tier storage vendors generally raise the MSRP excessively so that they can make it seem that they are giving you a great deal. They're not. Keep in mind that the media is the largest hardware cost in a storage system. The average storage system media street price after all discounts is minimally 3x the average street price for the same drive in a server.
- **The maintenance or subscription price are based on the discounted net price**
    - With a couple of exceptions, no. Maintenance and subscription pricing is calculated based on MSRP.
- **Adds and changes are tied to the same discounted price.**
    - Highly unlikely. Because you can't buy hardware, parts, or media from anyone else and expect it to work, you are a captive account. Like maintenance and subscription, adds and changes are sold at a much higher price than the initial discounted hardware purchase price.
- **Admin and user productivity are sunk costs.**
    - Definitively untrue. Productivity is directly related to project completions, time-to-market, time-to-revenues, morale, full-time employee (FTE) headcount, FTE turnover, customer turnover, and profits. Higher productivity equates to fewer headcount going forward, allocation of current FTE to more strategic tasks, reduced turnover and training costs, overall lower costs, higher revenues, and higher profits. Definitely not sunk costs.
    - The reason many CFOs and CTOs believe personnel are sunk costs is because they do not think what they pay for FTEs will change if productivity goes up. They don't consider morale, turnover, loss of knowledge, skills, and experience, and training costs. Nor do they account for the increased revenues from faster-time-to-market. It's a very short-sighted view.
- **Rack space or rack units (RU) consumed are insignificant to cost considerations.**
    - Data center fixed overhead is allocated per RU. Co-locations also charge per RU reserved and used. RU is very relevant to data storage costs.
- **Cost of all other supporting infrastructure is generally the same for all data storage.**
    - Not even close to true. Some need more block data storage systems to provide the necessary performance or scalability. That requires more switch ports, switches, transceivers, cables, cable management, conduit, etc. In addition, as previously discussed, many applications such as online transaction processing databases may need to be sharded hitting different block data storage systems. That means more application software licenses, subscriptions, or SaaS

rentals, plus more application server hardware, virtual machines, or containers, NICs, switch ports, cables, cable management, transceivers, conduit, etc., again.

- **Lack of automation or STaaS is irrelevant because our data storage admins know what they're doing.**
  - Possibly. But instead of the time consuming, labor-intensive, pedantic data storage administration, and management, or custom manually built scripts, they could be using their time for more strategic efforts. Efforts such as overall application performance tuning, application development, or continuous innovation.

Better Solution: **Calculate the total cost of ownership (TCO)**



- Purchase price (P)
- Total prem maintenance 3-5 yrs. (M)
- Or total STaas over 3-5 yrs.
- Add'l HW p + m or STaaS
- RUs consumed
- Ficed DC overhead/RU x RUs
  Power & Cooling & UPS

- App license consolidation savings
- App HW consolidation savings
- Productivity savings
- FTE hrly savings x cost/hr.
- FTE turnover/training savings
- Time-to-market unique revenues

The first 2 parts of TCO are "hard" and "soft" costs. Hard costs are directly measurable that include:

- Purchase price, subscription, or monthly STaaS fees;
- All other required hardware infrastructure - servers, NICs, adapters, switch ports, switches - purchase price, subscription, or as-a-service fees;
- RUs consumed multiplied by the data center fixed overhead allocated per RU; cables, transceivers, conduit; supporting the storage.

Soft costs are indirect costs that include:

- Personnel such as storage admins.
- Personnel churn.
- Training costs.
- Productivity costs - costs directly tied to application response times that exceed the Doherty Threshold. See Appendix A.

The third set of costs are known as opportunity costs. These are lost revenues as a result of inadequate storage performance. These include:

- Late versus early time-to-actionable-insights.
- Time-to-action.
- Time-to-market.
- Time-to-unique-revenues/profits.

Evaluating and Comparing Block Data Storage Systems or Software

Know thyself

As discussed in the introduction, it starts with knowing all of your needs, requirements, current and future hardware infrastructure. Below is an important application workload worksheet.

Block Data Storage Technical Requirements Worksheet

App Workload Name				
Requirements				
Latency				
IOPS				
Reads/writes/mix				
Today				
1 yr. from now				
2 yrs. from now				
3 yrs. from now				
Throughput - Bps				
Now				
1 yr. from now				
2 yrs. from now				
3 yrs. from now				
Capacity - TBs				
Now				
1 yr. from now				
2 yrs. from now				
3 yrs. from now				
RPO				
RTO				

Next up is making sure you know what your current infrastructure looks like in servers, networking, and current data storage systems or software.

IT infrastructure

Storage network	Select all that apply	Server Vendor	Server Model
Storage Network			
Ethernet 1Gbps			
Ethernet 10Gbps			
Ethernet 25Gbps			
Ethernet 40Gbps			
Ethernet 50Gbps			
Ethernet 100Gbps			
Ethernet 200Gbps			
IBA 10Gbps			
IBA 40Gbps			
IBA 56Gbps			
IBA 100Gbps			
IBA 200Gbps			
IBA 400Gbps			
FC 8Gbps			
FC 16Gbps			
FC 32Gbps			
FC 64Gbps			

Features/Functions/Capabilities

Unless the features, functions, and capabilities serve a specific purpose such as meeting your requirements or solving a problem, then they are only technology. It may be interesting while not particularly useful if it does not actually meet or exceed a requirement, solve a problem for you of some sort, or advance your organization in a meaningful way. If it does not do any of those things, it's a waste of time and money. Many data storage systems have features that are meant to differentiate them from their competitors, whether they're useful or not. There are many times said unique feature is there to just fix a product problem.

Here is a great example: several years ago, a major storage vendor used Microsoft Windows as part of its data storage OS. Then a major Windows malware attack occurred worldwide. But when it was cleared out of all of the infected servers, something kept reinfecting them. It was tracked down to the data storage system. The vendor had to install anti-virus software on all of their storage systems. They turned that negative into a marketing positive by saying "our data storage systems have anti-virus software, does yours?" Doesn't matter that no other data storage required it.

Caveats

Some vendors push deduplication and/or compression as a way to make their cost/TB appear lower. Deduplication and compression are only effective for some types of data such as backups, archives, or various types of unstructured data. They're not nearly as effective with most database data, video, audio, or encrypted data. In fact, many of the applications that produce this type of data have their own much more effective compression and deduplication built-in. And many encrypt the data before it's written to storage.

Take the Oracle Database as an example. The best case deduplication and compression of Oracle Database data is ≤ 2:1. Oracle's built-in hybrid columnar compression is 10:1 or higher in some cases. Data storage based deduplication and compression add latency on both writes and reads noticeably reducing performance. Just remember, they are primarily aimed at reducing capacity cost at the penalty of performance.

Another caveat is how vendors measure their performance. For latency, is it measured from the application server to the storage and back? Is it based on reading or writing? For IOPS, what is the payload - 4K blocks, 8K blocks, 16K blocks, etc. Are the IOPS sequential or random reads or writes, or a mix of random reads and writes? Throughput is measured in bytes per second (Bps). The key is how it's calculated. Is it calculated bidirectionally or unidirectionally? And for all of the performance measurements, are they calculated per workload, per node, or per system?

Because vendors' performance statistics tend to use different methodologies, or in some cases do not report any performance statistics, comparisons can be difficult. This needs to be normalized for effective and correct comparisons. The best way to normalize performance comparisons is to provide each of them with a benchmark or test that emulates your workloads. Have each of the vendors run it in their labs based on the configuration they want to sell to you.

Remember that published performance statistics will likely not resemble what you will see. A lot of that has to do with mixed workloads, inefficient data storage software, not enough nodes and lots of different demands happening concurrently.

Comparing Products and/or Services: Step 1

Start by creating general categories for performance, scalability, data protection (includes security), manageability (ease of use or simplicity), and cost. Then under each category, list all of the features, functions, and capabilities you require or think you need.

Next, mark all features, functions, and capabilities with one of three different classifications. Items marked with the first classification are the must haves. Failure to provide any one of the must haves is a deal breaker. Some good examples are specific performance or data protection capabilities. If an OLTP database workload needs 1 million IOPS, you can't accept block data storage that doesn't provide at the very least that much and probably more. The second classification indicates the important capabilities. These are perceived as borderline necessities; however, you can live without some of them. A common example is RTO. You may want to have RTOs (recovery time objectives) that are instantaneous. But can you live with a minute or two.

The third classification identifies the nice to haves. These are advantageous, have value, but are not necessarily a requirement. Yet they can be the difference maker.

Then take each data storage system or SDS you are evaluating and put them in their own column. Rank each feature, function, capability between 1 and 5, with 1 being the lowest and 5 being the highest. You can also optionally put weights on them. This adds a bit of complexity that is not necessary because of the three classifications. You should be looking for 100% of the 'must haves', 60-85% of the 'important', and 25-50% of the 'luxuries'.

See example spreadsheet below.

Block Data Storage Product/Service Feature, Function, Capability Comparison

Vendor										
Product / Service										
Must Have (M), Important (I), Luxuries (L)	M/L/I		M/L/I		M/L/I		M/L/I		M/L/I	
Performance										
Best latency										
Max sequential read IOPS/system										
Max 30% writes, 70% reads random IOPS/workload										
Max sequential read IOPS/server, or node										
Max 30% writes, 70% reads random IOPS/controller, server, or node										
I/O load basis - 4K, 8K etc.										
Max total throughput										
Max throughput/workload										
Max throughput/controller, server, or node										
Scalability in TBs										
Max SSD raw capacity										
Max SSD raw capacity/node										
Max HDD raw capacity										
Max HDD raw capacity/node										
Data Protection										
RAID capabilities										
Erasure Coding										
Across drives										
Across nodes										
Triple copy mirroring										
Snapshots										
Max per volume										
Max per system										
Lowest RPOs										
Lowest RTOs										
Immutability										
2FA										
Anomaly detection										
End-to-end encryption										
Encryption method										
Role based administration										
Manageability - Ease of Use										
Intuitive GUI										
HTML5										
CLI and/or RestFUL API										
Threshhold alerts/alarms										
Can be managed remotely										
Non-dsiruptive patching & upgrades										
Non-dsiruptive node or drive replacement										
Non-dsiruptive tech refresh										
Non-disruptive drive replace										

If none of the vendors meet your criteria, you have 2 choices. Either look for a data storage vendor that manages to meet the criteria or reevaluate and modify the criteria.

TCO

TCO is much more than simply the purchase price plus maintenance, leasing/financing, or STaaS. There are many costs that really need to be counted. There also needs to be accounting for application software license or subscription consolidation, and hardware consolidation cost savings because of faster storage providing faster application response times. Those faster application response times also convert into user productivity gains, FTE time savings now available for other more strategic tasks, reduced FTE turnover plus their training, and unique revenue gains.

All of these positive credits against cost are considered 'soft' by many CFOs. They will disregard or not include those credits. That's a huge mistake in that they can and commonly do dwarf the costs. There are significant and dramatic differences between vendors. Here's how to calculate those credits.

When the application is waiting for the data storage, faster data storage translates into faster application response times. Faster response times enable application and database consolidation. Application consolidation savings are the easiest to calculate. Applications are typically licensed or subscribed on an instance, socket, or core basis. This is definitely true for databases. Even open-source software support is based per instance. The applications that charge on a per seat basis tend to be SaaS, and per seat licensing will not experience any license savings from consolidation. There will still be hardware consolidation savings in both cases because the application software needs to run on fewer virtual or physical servers. The credits then come from both reduced license, maintenance, or SaaS fees.

Productivity savings credit calculations are also straightforward. See Appendix A to calculate productivity cost savings.

Calculating FTE time cost savings requires knowing the average block data storage admin FTE annual cost. It will vary by region. However, a good rule of thumb is approximately $150K USD[3] including all benefits. Next determine the number of hours saved because of automation, intuitive simplicity, and/or STaaS. This will require some research, talking to the current in-house block data storage admins. Then estimating the total hours saved per week, month, and year. Multiply those hours by the average FTE cost per hour: hours per year = 252[4] working days * 8 = 2016 $150,000/2,016 = $74.40/hr.

There is also the turnover, new hire, and training savings. This too is tied to freeing up the admin's time. It requires a bit of estimating how improved morale will decrease turnover. Starts with estimating the cost of hiring a new admin. That cost is much higher when executive recruiters are used. Then there's the training cost for a new block data storage admin. It should include any vendor system training and any other tools they need to be trained up to speed. Multiply that cost times the number of admin FTE turnover saved per year.

The last bit of revenue calculation is unique revenue gains as a result of faster application response times. This is the most difficult to calculate. That does not mean it should not be calculated. It definitely should. It means looking at the new product or service revenue projections based on time-to-market. Determining how much of that is accelerated because of productivity gains from faster application response times. Then using the projected growth curve from the new starting point. Then compare the 3-year and 5-year revenues to the original projected revenue curve. The differences from the new to the old are a net gain.

These credits against cost are all based on the different block data storage having different

3 This is an average in the USA. It will vary location. Different regions and countries will have different average FTE costs.

4 The number of working days will vary by region. 252 working days is applicable to the USA.

performance both now and later at scale. If performance was identical, then the credits would not matter. Performance is nowhere near the same.

Total Cost of Ownership (TCO)

Vendor				
Product / Service				
Debits				
Purchase price (P)				
Total prem maint 3-5 yrs (M)				
Total STaaS over 3-5 yrs				
Add'l HW p + m or aaS				
RUs consumed				
Fixed DC overhead/RU x RUs				
Power, cooling, & UPS				
Total				
Credits				
App license consolidation savings				
App HW consolidation savings				
Productivity savings				
FTE hourly savings x cost/hr.				
FTE turnover savings				
Time-to-market unique revenues				
Total				
Net Totals				

Many block data storage will not reduce FTE time, improve user productivity, or increase unique revenues because they do not accelerate application response times in a material way.

Recommendations

Choosing the best block data storage for your organization will always come down to your needs now and in the future. But do not sacrifice the future for the immediate now as too frequently happens. A specific block data storage might meet your needs upfront but can't meet where those needs may grow, especially in performance. Lower cost upfront does not mean it's lower cost over 3 to 5 years. You can meet your needs now as well as in the future with a little effort and planning on your part. It will pay off in a very large way.

Keeping that in mind, here are the block data storage capabilities you **should** be looking for to solve block data storage issues and problems and make your life much easier.

1. Best block data storage type

- **Scale-out - because it solves several difficult problems.**
    - Eliminates the limited data storage controller/node/server problem.
    - Provides highly scalable performance and capacity.
    - Delivers non-disruptive patching, upgrading, and tech refresh.
    - **Examples**: StorPool, Nvidia-Excelero, Dell-PowerFlex, iXsystems-TrueNAS, NetApp-Solidfire

2. Most flexible block data storage implementation - SDS

- **Radically reduces the cost of hardware.**
    - COTS servers and you can leverage current server vendors.
    - Enables faster adoption of faster CPUs, networks, and media for competitive advantages.
    - Media is more than 67% less costly.

3. Ideal performance

- **Very low latency and high IOPS**
    - Empowers mission-critical or business-critical application sub second response times.
    - These applications are not waiting on the data storage.
- **High throughput**
    - Enables analytics, machine learning (ML), deep machine learning (DML), process AI (pAI), or generative AI (gAI), for faster results.
- **Extremely efficient capacity utilization that does not impact performance.**
    - Aim for 90% or better without performance degradation.
- **Snapshots that do not degrade performance while being taken.**
- **Drive or data rebuilds that do not degrade performance while occurring.**

4. Essential performance and capacity scalability

- **A good rule of thumb for future performance scalability needs is the 2x exponential factor.**
    - Take the performance requirements required in year 1 and double it for year 2.
    - Take the performance requirements required in year 2 and double it for year 3.
    - Take the performance requirements required in year 3 and double it for year 4.
    - Is that more than what's required? Possibly, but just as likely not.
- **Historical capacity planning rule of thumb, take the 3-year estimate and double it, no longer valid.**
- **Data storage is doubling approximately every 2 years and accelerating.**

    - Because of ML, DML, pAI, gAI, analytics, data lakes, data lake houses, IoT, and more.
    - Plan 4-10x the current amount of data stored for the next 3 to 5 years.

5. Better Data Protection

- **Enough snapshots per volume to meet mission-critical and business-critical application RPOs**
    - Now and over the next 3-5 years.
    - Expect more workloads requiring a minimum RPO as time goes on.
- **Immutable snapshots to prevent unauthorized deletions or modifications.**
- **Two factor authentication (2FA) to prevent unauthorized access or changes.**
- **Erasure coding across data storage drives and nodes.**
    - As long as it does not negatively impact performance.
    - Rebuilds data, not drives, faster rebuilds, reduces risk, and reduces capacity needs.
    - Minimally, RAID 5, 6, 1, 0, 10, 50, 60, although erasure coding should be preferred.
- **Triple-copy mirroring or more between storage nodes.**
    - Delivering continuous data access for multiple hardware failure types.
- **End-to-end data encryption.**
    - Prevents unauthorized access of data in-flight or at-rest.

6. Simpler manageability

- **Non-disruptive patching, upgrades, drive replacements, software changes, or tech refresh.**
    - Disruptions have to be scheduled for non-production hours in a 7x24x365 world, are on a tight schedule, pressure-filled and are often error prone requiring do overs.
    - Disruptive patching delays implementation of vulnerability patches.
    - Non-disruptions can be done online during production hours without taking an outage.
- **Intuitive GUI**
    - Enables novice data storage admin to use and figure out everything with little to no training.
- **CLI or RESTful API**
    - For integration with other data center management tools.
- **Threshold alerts and alarms**
    - Optional predetermined actions such as removing a drive from a volume based on errors.
    - Useful in dealing with problems promptly.

7. High degree of Automation

- **More automation equals fewer human errors, less required expertise, reduced admin workloads.**
- **Reduces data storage admin turnover.**
- **Saves time and money.**

8. Lowest TCO/performance

- **Make sure you're looking at the lowest TCO and not the lowest price**
    - TCO/IOPS and TCO/Bps.

9. Choice of payment methods

- **STaaS consumption model of pay for what you use**
    - Reduces risk, management, and overpaying upfront.
    - But it can cost more in the long run.
    - Per Gartner, STaaS can decrease IT spend by up to 40%.
- **Purchase upfront + warranty + premium maintenance**
    - Traditional method.
    - Risk of overbuying upfront to take advantage of upfront discount.
    - Premium maintenance based on MSRP not net price.

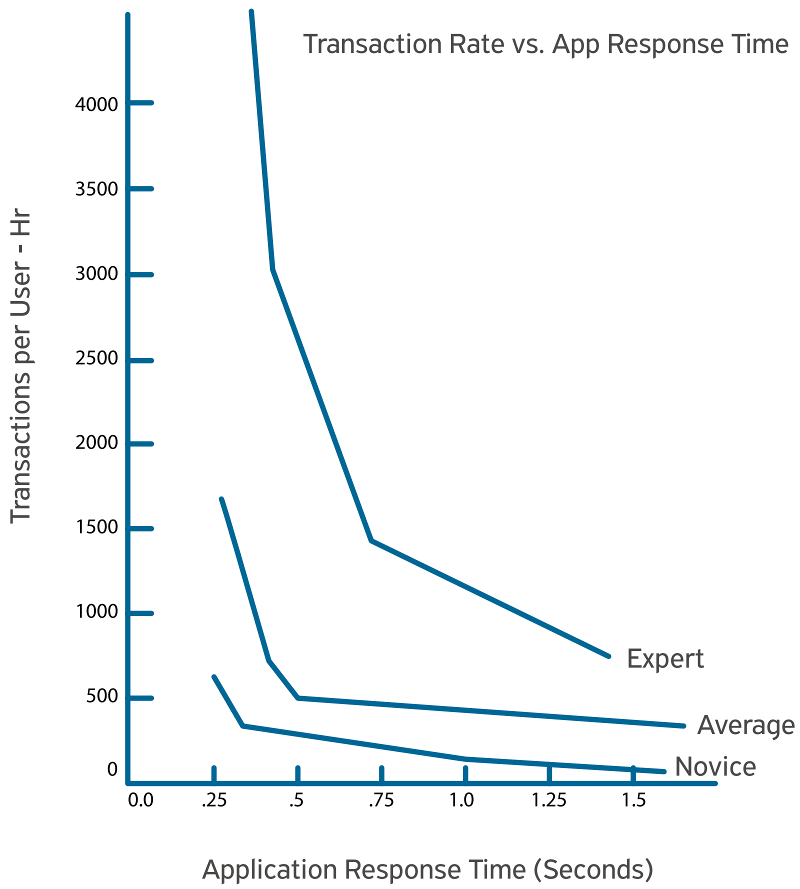
Final Thoughts

Choosing the right block data storage is non-trivial. It will have repercussions for years and greatly affect your organization's ability to compete. Do not make your buying decision lightly or cavalierly. It is too important.

Appendix A: Rapid Response Time Economic Value

Calculating performance productivity impact:

- **Performance has a substantial and measurable impact on productivity.**
    - Response time has a direct correlation on user productivity, quality-of-work, and time-to-market.
    - It was determined that the maximum application response time before user productivity declines precipitously is 3 seconds. Anything over 2 second response times caused user attention to wander.
    - Application response times that are less than 3 seconds promptly increase user productivity, quality-of-work, and time-to-market.
    - Reducing response time to ~ .3 seconds more than doubles productivity versus 2 seconds. Productivity gains are substantially greater depending on the user's level of expertise.



• Faster response times mean shortened project schedules and higher work quality.

• ≤ .4 seconds equates into what is called the Doherty Threshold. The Doherty Threshold is when response time becomes addictive whereas > .4 seconds users' attentions begin to stray and productivity begins to decrease rapidly.

Application Response Time (Sec)	Transactions per Hr	Task Time (Min)	Time Saved per Task (Min)	Time Saved per Day
3	180	60	-	-
2	208	51.9	8.1	1h/4m/48s
1	252	42.9	17.1	2h/16m/48s
0.6	279	37.7	22.3	2h/58m/24s
0.3	371	29.1	30.9	4h/7m/12s

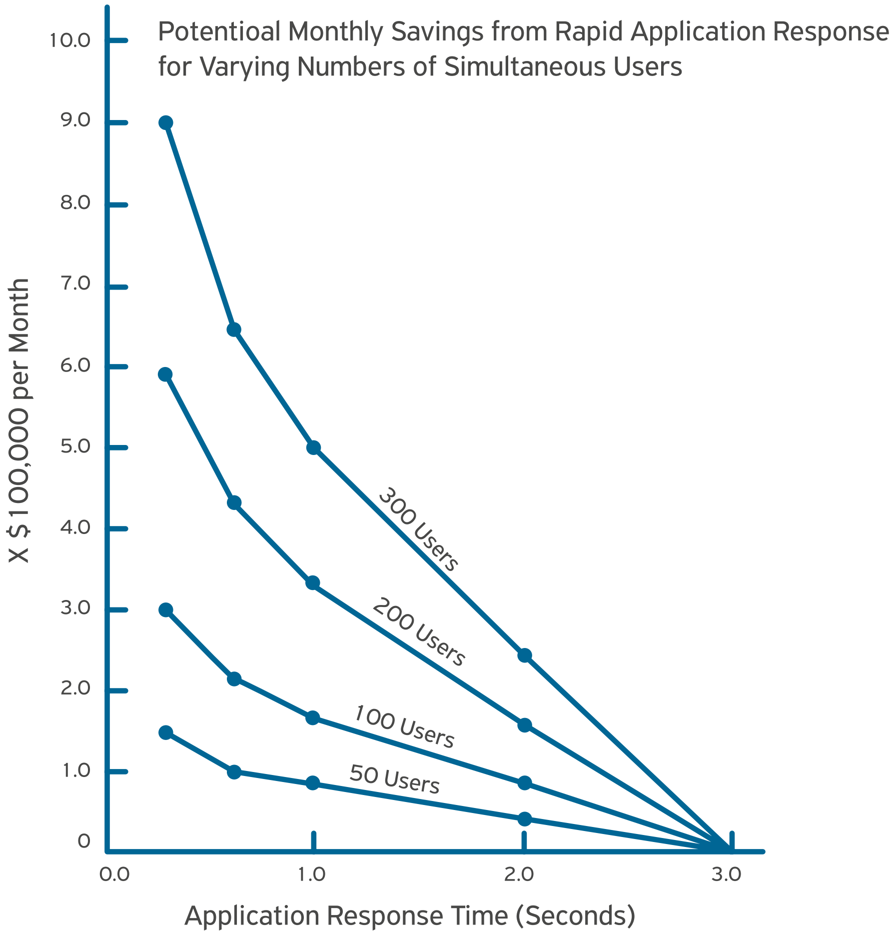
• Determine application response times for each service under consideration.

• Compare productivity rates.

• Divide FTE costs by productivity to calculate FTE cost per transaction.

• One alternative is to compare the time required to complete a defined set number of transactions.

• Multiply the time saved by FTE average hourly cost.



Time-to-market revenue acceleration increases top line revenues and bottom line profits

- **Based on current schedules estimate the following:**
  - Amount of revenue for each week or month schedule is moved up.
  - Project how much time the reduced application response time performance will accelerate the time-to-market. This can be derived from the increase in productivity based on application response time. If the developers can more than double their productivity, they can more than cut in half the amount of time to complete their project.
  - Apply the projected market growth rate to that revenue for a set period, anywhere from 1 - 10 years. Compare the total revenues to what it would have been had the schedule not been accelerated. The differences are the unique gains. If the database cloud service delays time to market, then the differences are the unrecoverable losses.
  - Example from a large microchip manufacturer:
    - By accelerating delivery of their chip to market by one quarter they were able to realize unique revenues > than $100 million upfront and five times that amount over 3 years.

StorPool
DISTRIBUTED STORAGE

The Ultra-Fast & Effortless Block Storage Software Platform

Talk to an Expert Today

www.storpool.com

+1 415 670 9320 (USA)
+44 (0) 88 834 6465 (Bulgaria)

info@storpool.com