Technical Validation

Pure Storage FlashBlade: Unified Fast File and Object (UFFO) Platform

Simple Management and Multi-dimensional Performance Enable Consolidation

By Kerry Dolan, Senior IT Validation Analyst

March 2021

Contents

Introduction 3
Background 3
Solution: Pure Storage Unified Fast File and Object Storage (UFFO) 4
ESG Technical Validation 5
Simplicity 5
ESG Testing 5
High Availability and Multi-dimensional Performance 11
ESG Testing 11
The Bigger Truth 14

ESG Technical Validations

The goal of ESG Technical Validations is to educate IT professionals about information technology solutions for companies of all types and sizes. ESG Technical Validations are not meant to replace the evaluation process that should be conducted before making purchasing decisions, but rather to provide insight into these emerging technologies. Our objectives are to explore some of the more valuable features and functions of IT solutions, show how they can be used to solve real customer problems, and identify any areas needing improvement. The ESG Validation Team's expert third-party perspective is based on our own hands-on testing as well as on interviews with customers who use these products in production environments.

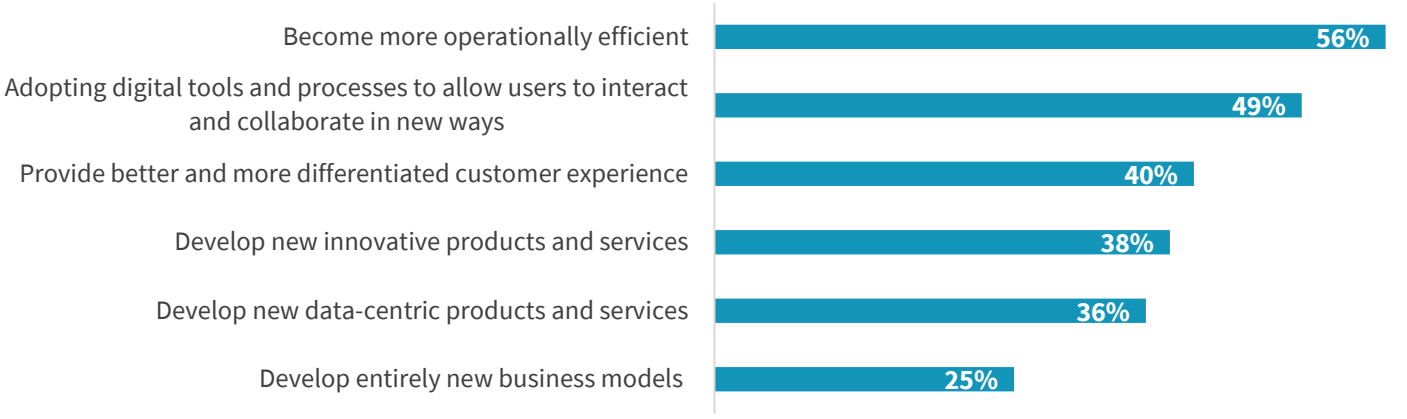
Introduction

This report documents ESG's validation of Pure Storage FlashBlade as a platform for consolidating both file and object data. The report focuses on how FlashBlade's ease of use and multi-dimensional performance deliver what organizations need to consolidate file and object workloads, increasing efficiency, productivity, and cost savings.

Background

Operational efficiency is the most-cited objective for digital transformation according to ESG research (see Figure 1).[1] Organizations are collecting, storing, and analyzing more data as they leverage artificial intelligence (AI), machine learning (ML), data analytics, the Internet of Things (IoT), and other modern technologies. However, most organizations silo their applications based on data types, making their infrastructures inefficient, difficult to manage, complex, and costly.

Figure 1. Efficiency Tops Digital Transformation Initiatives



Source: Enterprise Strategy Group

Unstructured data (e.g., files, video, audio, logs, sensor data, etc.) can be managed and stored as files or as objects that are accessed by unique identifiers. These data types are proliferating on a massive scale and are valuable because of the real-time insights they can provide. When ESG asked respondents what workloads they believe will be responsible for the most on-premises and cloud storage growth over the next 24 months, IoT and big data/data lake repositories for unstructured data were the most-cited responses. The same research found that, between 2017 and 2019, on-premises object storage grew by more than 3x.[2]

Traditional storage has typically not been used for both files and objects, or even for different types of file applications. The challenges that have hindered consolidation include:

- The need for separate NAS silos optimized for specific workloads, such as for applications using many small files versus those using fewer, larger files.
- Deployment of separate systems by protocol, one for file (NFS and SMB) and another for object (S3).

[1] Source: ESG Research Report, *2021 Technology Spending Intentions Survey*, January 2021.
[2] Source: ESG Research Report, *Data Storage Trends in an Increasingly Hybrid Cloud World*, March 2020.

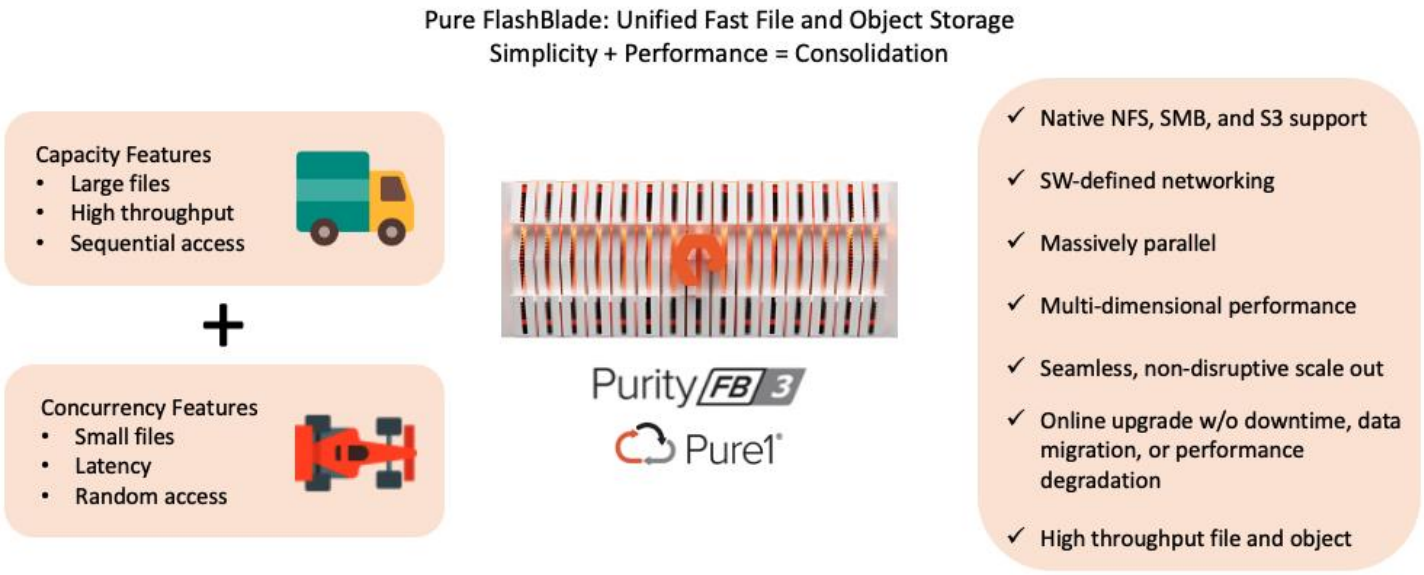
- Complexity of storage management within and across silos.
- Inability to scale performance with capacity for different types of NAS workloads, such as those with large numbers of files or objects.
- Fork-lift upgrades and a-la-carte feature licensing.

As file and object data are increasingly analyzed to deliver insights and inform business decisions, they must be efficiently stored, easy to access, and highly available, all with high performance. While object storage was traditionally used as a long-term archive and did not need speed, today these objects are in high demand for analysis and must be accessible on a high-performance platform. Modern infrastructures must be able to handle varying workloads that use huge volumes of large and small files and objects, often concurrently.

Solution: Pure Storage Unified Fast File and Object Storage (UFFO)

Pure Storage UFFO, centered on FlashBlade technology that is powered by the Purity//FB operating system, offers a native flash-based platform with native file and object storage (not a gateway). This scale-out platform leverages flash to provide performance and efficiency so customers can consolidate file and object workloads with confidence.

Figure 2. Pure FlashBlade Unified Fast File and Object Storage



Source: Enterprise Strategy Group

Performance + Simplicity Enables Consolidation

What makes Pure FlashBlade the right UFFO platform for consolidation? First, it is an all-flash array; that alone delivers high performance, which may not be new for file data but is definitely exceptional for object storage. FlashBlade's highly parallelized architecture delivers multi-dimensional performance, with scalable capacity and file/object concurrency. This performance lets customers consolidate file and object workloads to reduce floorspace, power, and cooling costs. Customers can count on performance for workloads with different I/O patterns, whether your applications use small or large files, small or large objects, or massive volumes of files or objects. As a result, customers can combine their analytics, AI/ML, high-performance computing, and other applications on the same platform. Instead of duplicating data for different

siloed applications, organizations can leverage one set of data for greater efficiency and reduced cost. Performance scales predictably, as every blade includes a full set of components, including processors, networking, DRAM, PCIe connectors, NAND storage, and the Purity//FB operating environment.

Pure FlashBlade has simplicity designed in. It is a scale-out system that dynamically load balances across all blades without IT intervention; in addition, software-defined networking is integrated in FlashBlade, vastly simplifying network management. Customers can scale out the fabric non-disruptively. The modular, software-defined architecture includes upgradable blades (for capacity and density) and fast, online upgrades without downtime, data migration, or performance degradation.

Organizations can easily scale on-demand and non-disruptively, and there is no manual tuning; while other arrays require IT to manage node counts, pools, protection schemas, background jobs, etc., the Purity//FB operating environment takes care of that. FlashBlade is self-healing and auto-scaling, and redundant components ensure high availability. In addition, Pure1 provides the opportunity to centrally monitor a fleet of arrays from a single console using a web browser or mobile device.

Purity//FB also includes essential capabilities such as file and object replication for disaster recovery, cloud mobility, and secondary use. File system rollback also enables fast file system recovery from snapshots. Other features include inline data compression and thin provisioning, erasure coding, snapshots, and always-on encryption.

ESG Technical Validation

ESG viewed remote FlashBlade UFFO demonstrations with FlashBlades located in Dallas, TX and New York, NY. The demonstrations focused on the management simplicity and multi-dimensional performance that make FlashBlade a great platform for consolidation.

Simplicity

Pure FlashBlade was built from the ground up with simplicity and ease of management in mind. The user interface (UI) is simple and intuitive, and because the Purity operating environment manages most storage-related tasks, IT has very little tuning to do. Our testing showed how easy it was to create a file system and an object bucket, and to initiate replication.

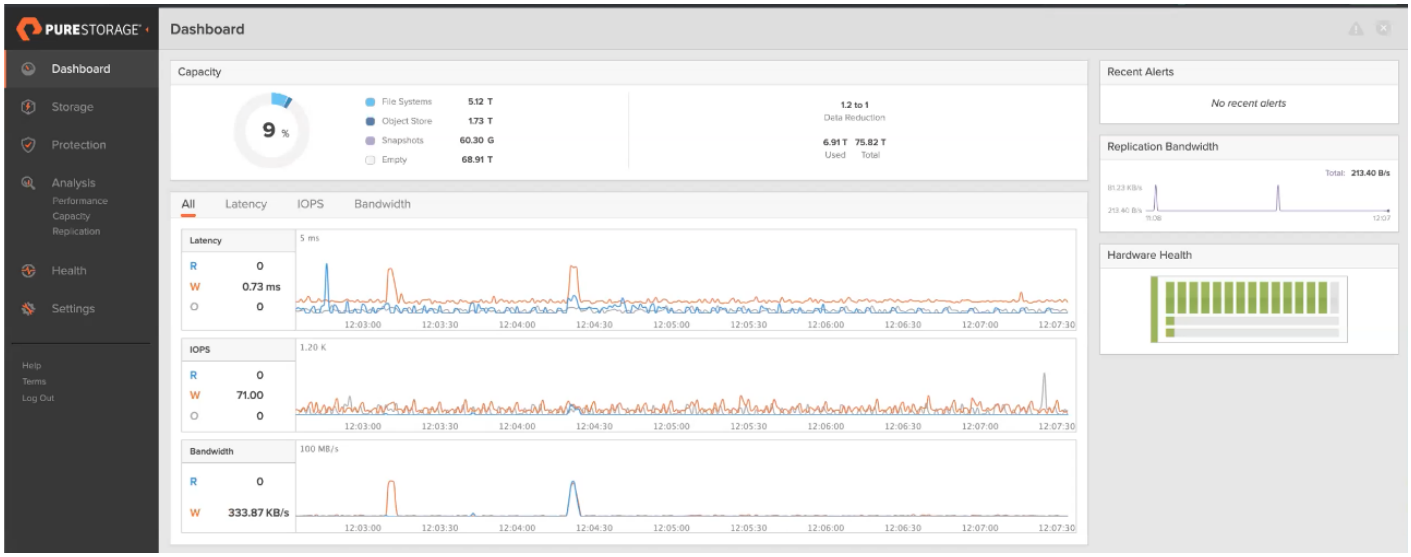
ESG Testing

ESG started with a look at the dashboard. We noted how simple and clean the UI looked, with just the most important information displayed:

- A capacity ring, showing file systems, object stores, and snapshots, plus the data reduction ratio and amount of capacity used.
- Current read/write latency, IOPS, and bandwidth.
- Recent alerts.
- Replication bandwidth.
- A color-coded hardware health display showing the chassis and each blade.

Tabs on the left allow users to look more closely at storage, protection, analysis (performance, capacity, and replication), and health details, including inspection of individual components. System, network, user, and security details and settings are also available. We created a new network interface by simply naming it, adding the IP address, and assigning a service, either data or replication.

Figure 3. Clean, Simple FlashBlade UI

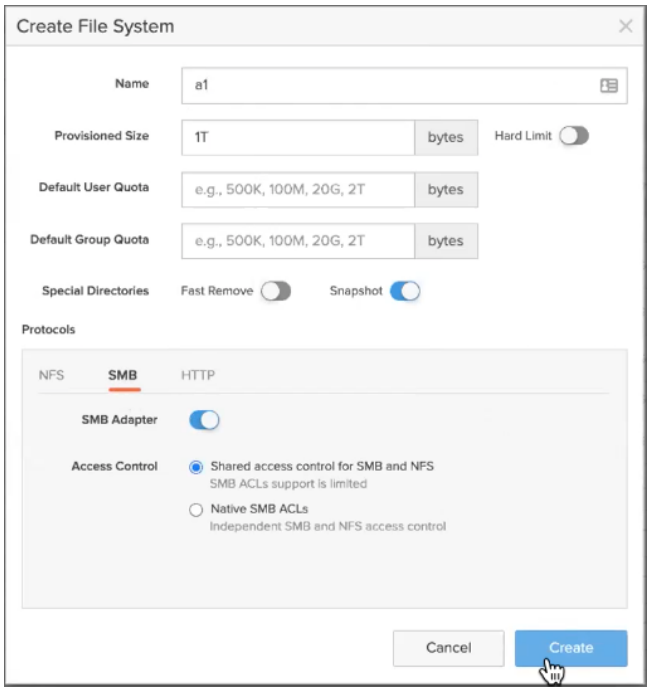


Source: Enterprise Strategy Group

File Storage

Next, we created a file system in less than a minute. From the ***Storage*** tab we could view array, file system, and object storage details, including counts, policies, network connections, and S3 target connections. We clicked ***File Systems***, clicked the **+** sign, named the new file system *a1*, provisioned 1 TB, and selected the SMB adapter. That was it; no additional configuration was needed, such as RAID groups or metadata pools. Quotas are optional and configurable.

Figure 4. Create File System

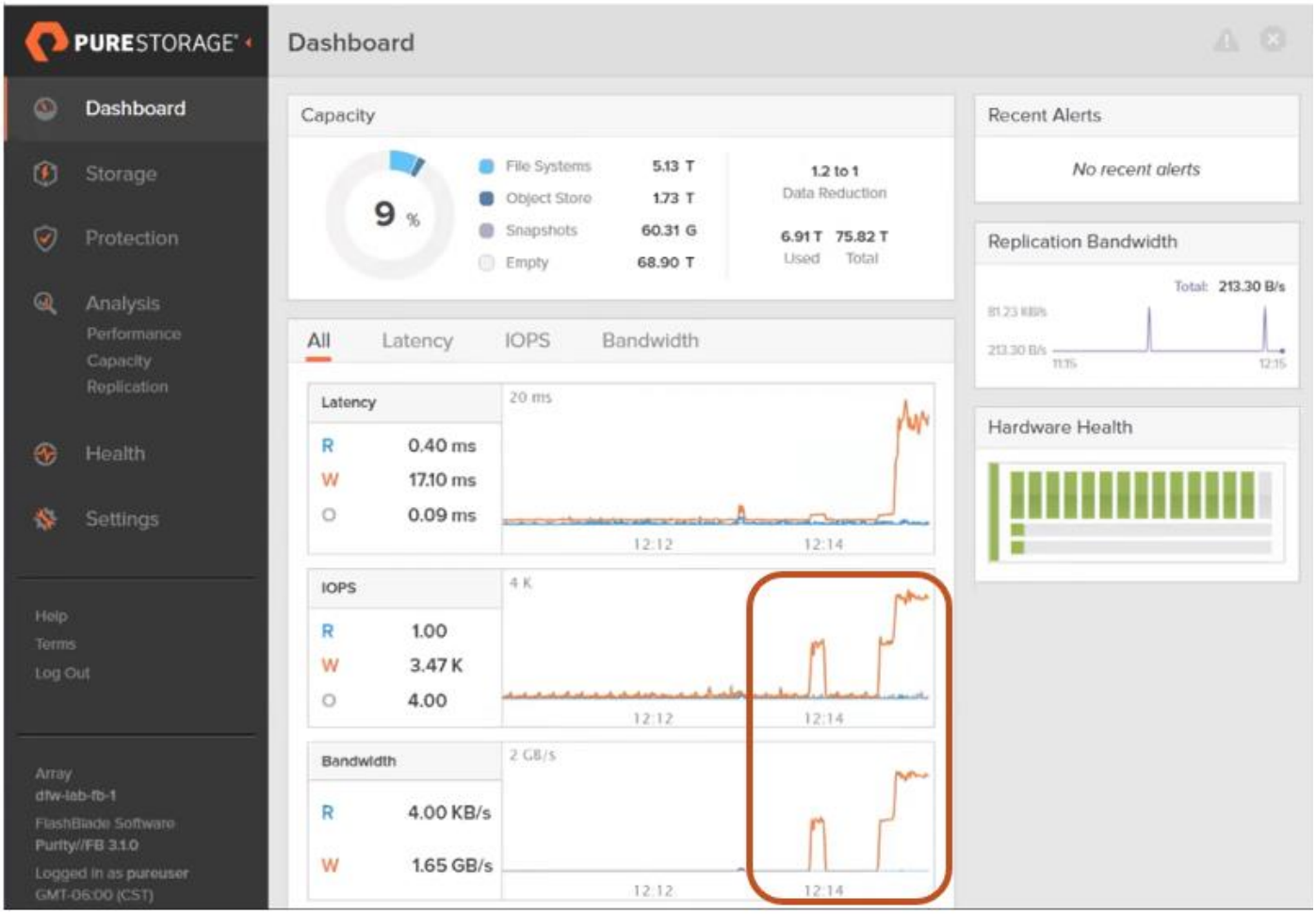


Source: Enterprise Strategy Group

Next, we navigated to the new file system from a Windows virtual machine (VM) and dragged a notes.txt file to it. We connected to a Linux VM using SSH, mounted the same filesystem, and validated activity by running a simple script; we could view that activity on the FlashBlade dashboard.

To demonstrate how FlashBlade scales, we wrote to the Dallas file system, starting with a CentOS VM and then adding an Ubuntu VM. FlashBlade is built for parallelism, and we noted the linear scale when the second workload was added. Figure 5 shows the IOPS and throughput scaling as each workload was added.

Figure 5. Linear Scale as Workloads are Added

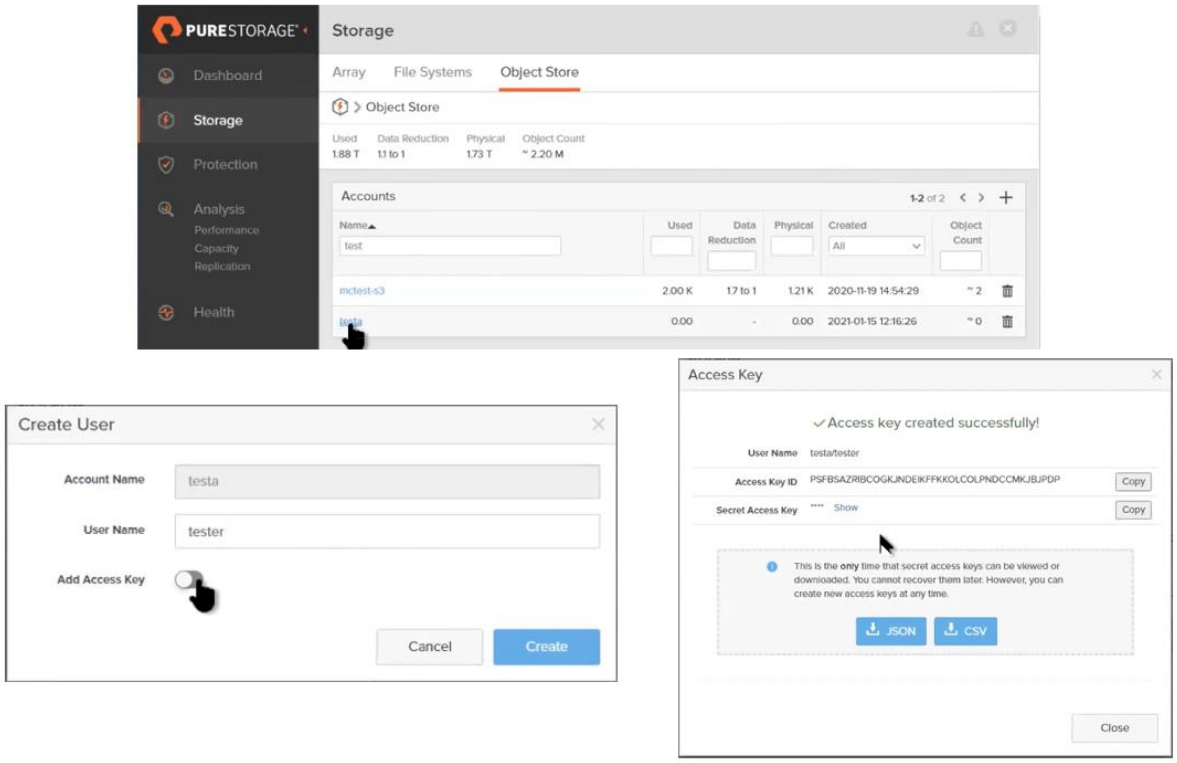


Source: Enterprise Strategy Group

Object Storage

Next, we explored the object side of FlashBlade UFFO, noting how fast and easy it was to create a new account, new user, and object bucket. From ***Storage/Object Store***, we clicked the **+** sign, gave the new account the name *testa,* and clicked ***Create***. To create a new user, we selected the account name (*testa*) and provided a username (*tester*). Object storage is authenticated via access and secret keys, so we created those and downloaded them to JSON since they cannot be stored persistently for security reasons.

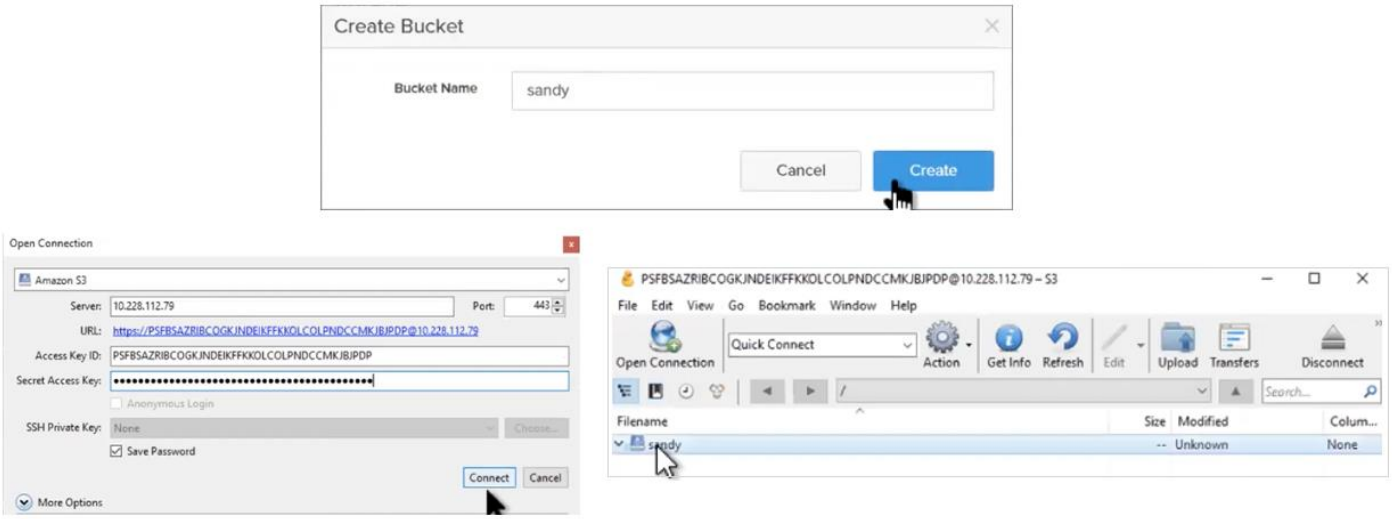
Figure 6. Create Object Storage Account/User



Source: Enterprise Strategy Group

Creating an object bucket was equally easy; we simply clicked **+** and named a new bucket *sandy*. To access this bucket, we opened Cyberduck (an open-source client supporting the Amazon S3 protocol) and added a new connection using the network interface previously created. After inserting the access and secret keys, we connected to the *sandy* bucket and executed a PUT by dragging an ISO image onto the *sandy* bucket.

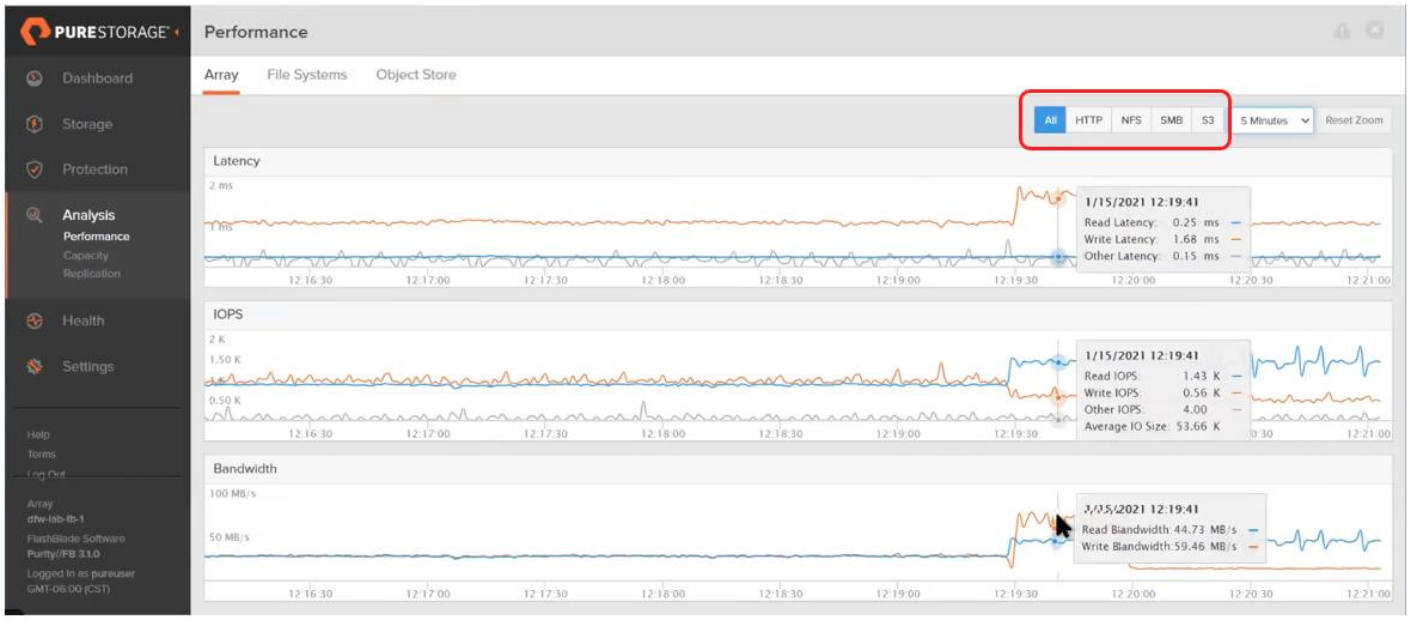
Figure 7. Create Object Storage Bucket



Source: Enterprise Strategy Group

In addition to creating users and accounts and getting access to file systems or object buckets easily, Pure FlashBlade shows performance for all protocols from a single interface. We clicked on ***Analysis/Performance*** and could view dynamic time frames of performance, showing read/write latency, IOPS, and throughput for all protocols: HTTP, NFS, SMB, and S3. We could also view performance by individual file system or object bucket.

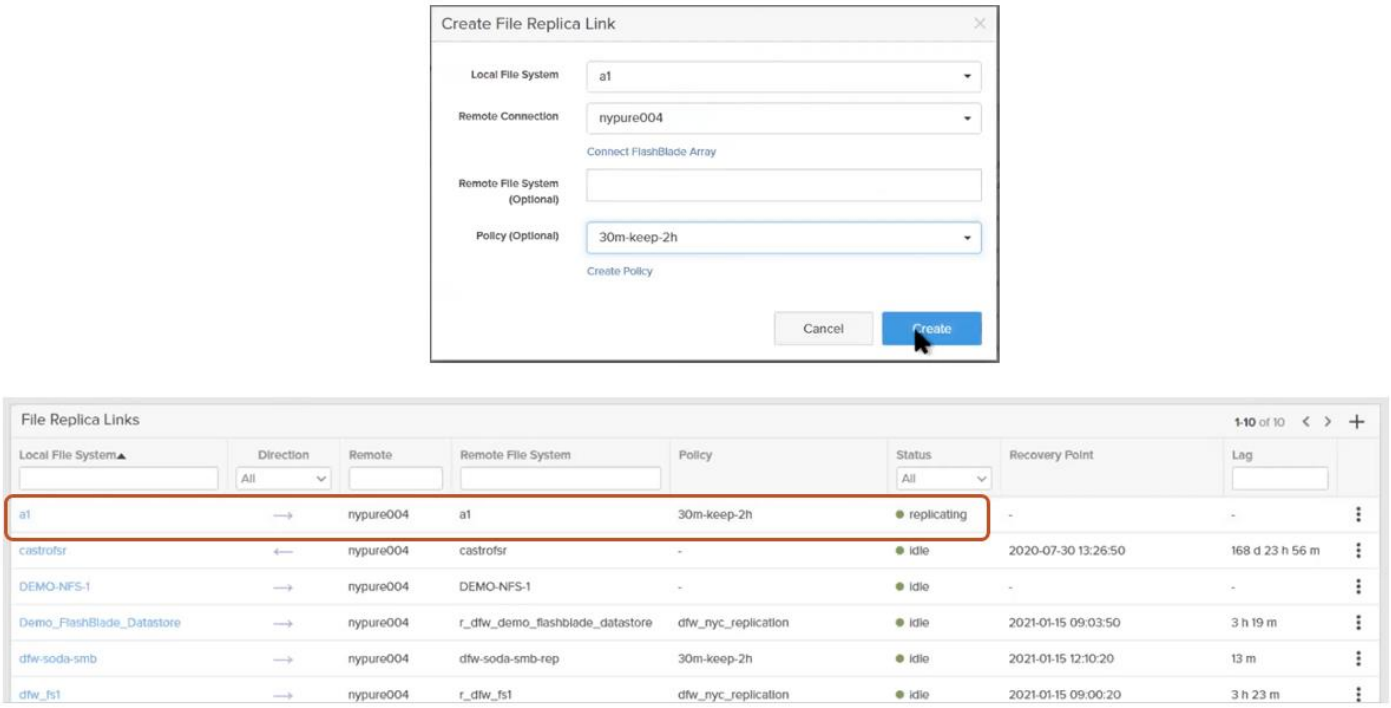
Figure 8. Multi-protocol Performance Monitoring



Source: Enterprise Strategy Group

Finally, we demonstrated the ease of configuring replication between the Dallas and New York FlashBlade systems. The systems were connected by simply generating a secret key on the remote system, pointing to a management IP address, and authenticating with the secret key. From the ***Protection*** tab, we selected ***File Replica Link*** (***Object Replica Link*** was also an option) and inserted the local file system name (*a1*) from which to replicate and the remote array (*nypure004*), with an option of inserting a remote file system as well. We selected a replication policy for scheduling and retention time and clicked ***Create***. Once that was done, the ***File Replica Links*** page showed that connection with an arrow pointing in the direction of replication; FlashBlade created a snapshot and began replicating. Once complete, the UI displayed the date and time of the last recovery points.

Figure 9. Initiating Replication



Source: Enterprise Strategy Group

Why This Matters

Storage management complexity, particularly when dealing with application silos, is a major cause of inefficiency and cost.

ESG validated the ease of executing storage tasks for both file and object data using the same interface on FlashBlade. It was simple and fast to create a file system and write to it; it was equally simple and fast to create an object storage account, user, and S3 bucket, and write to that. The UI was simple and clean, and we could view performance details for all file and object protocols from the same screen. FlashBlade's simplicity relieves IT administrators from tasks that take time and effort on traditional storage systems.

High Availability and Multi-dimensional Performance

ESG viewed additional demonstrations showing how FlashBlade serves NFS, SMB, and S3 workloads at the same time, how performance is maintained during blade evacuation and addition, and how it supports high object and file counts and concurrency.

ESG Testing

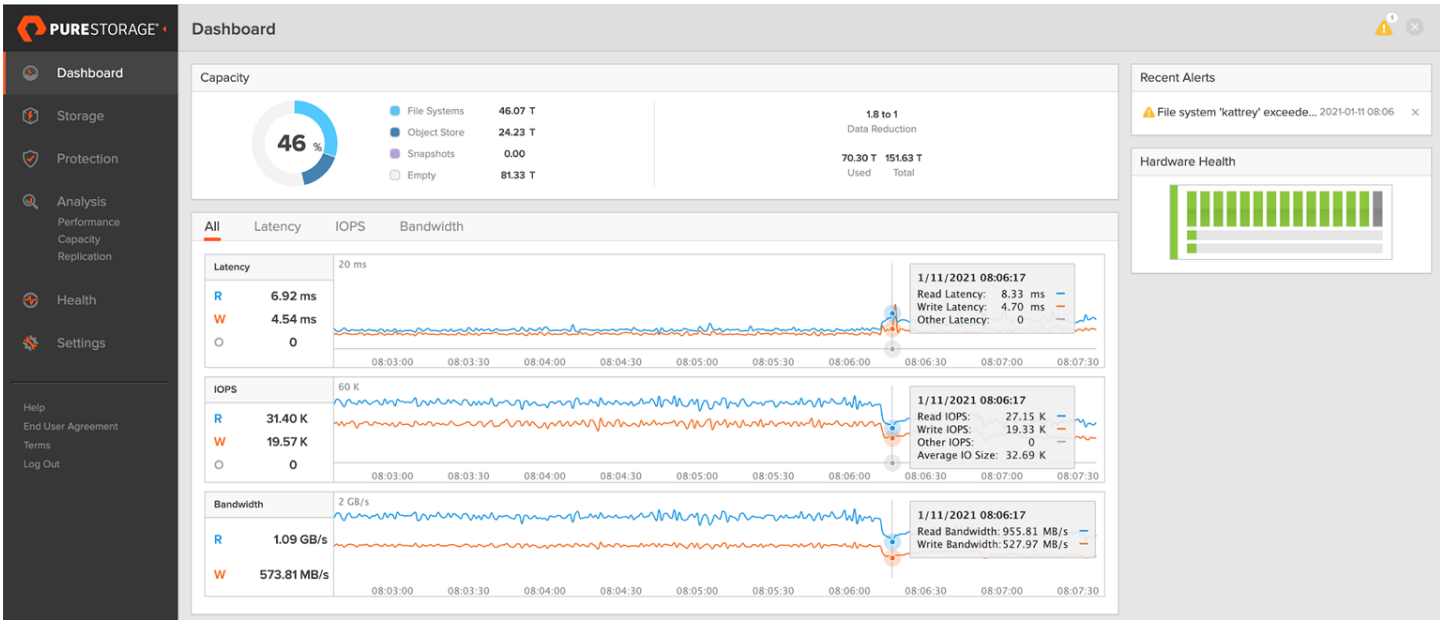
First, we viewed results of Pure testing during blade evacuation and addition, demonstrating how gracefully FlashBlade handles these changes, maintaining performance despite a heavy load. This is the kind of activity that occurs when replacing blades to increase density, for example. This test demonstrated the non-disruptive upgradability, ability to scale, and ability to consolidate multiple file and object workloads simultaneously. The test bed included 14 Dell R720 bare metal servers—four S3-connected, five NFS-connected, and five SMB-connected—each with 2 x 40GbE ports. FIO was used to generate 100% random, mixed read and write I/O with average I/O sizes, across all three protocols.[3] Hosts were connected to a Pure FlashBlade running Purity//FB 3.2, with 15 x 17TB drives and 4 x 40GbE ports.

With the workloads running full bore, we ran a command to remove Blade 15 from the cluster, requiring the system to move shards among blades. While performance dipped slightly, parity protection remained at 100%, and all three workloads continued uninterrupted. Next, we started evacuation of Blade 14 with the same result: a slight overall performance hit but no interruption of any workload. Once the blades were removed from the cluster, the SMB jobs halted due to the stateful nature of the SMB protocol, and we restarted them manually.

Next, we manually reintroduced Blades 14 and 15; SMB jobs disconnected, as expected, and were restarted, while NFS and S3 jobs experienced only a slight pause. Once the two blades had been re-added, and FIO workloads continued, FlashBlade continued data and parity shard rebalancing in the background with only slightly diminished performance; once rebalancing was complete, performance returned to initial levels. Figure 10 shows workloads continuing with only a minor performance dip during Blade 15 evacuation.

[3] The FIO workloads were all 100% random I/O. The workload for NFS had a 32KB block size, 70% read/30% write; the SMB workload had a 16KB block size, 30% read/70% write; and the S3 workload had a 64KB block size, 70% read/30% write. This test was not designed for maximum throughput or hero numbers.

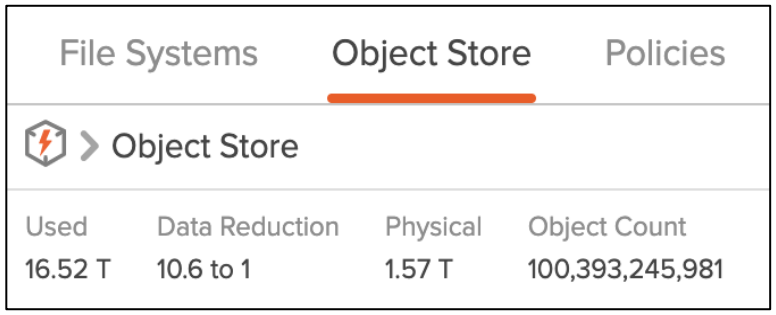
Figure 10. Performance During Blade Evacuation



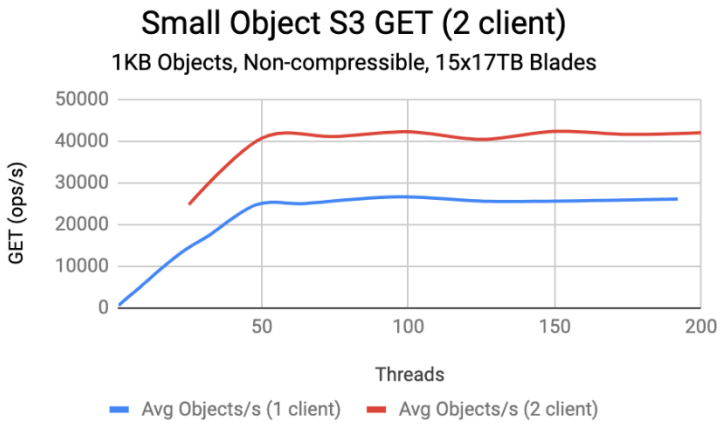
Source: Enterprise Strategy Group

High Object Count and Concurrency

Next, we reviewed results demonstrating high object count and concurrency. FlashBlade can support billions of objects because capacity is efficiently utilized across the cluster and there is no minimum block size. We viewed the UI of a Pure FlashBlade showing 1 billion (non-compressible) 1KB objects, consuming 1 TB of capacity; in contrast, arrays that have a minimum 4KB block size would need four times that much capacity for the same number of objects. We also viewed a UI showing 100 billion + objects with data reduction, showing consumed capacity of 16.5 TB logical but only 1.57 TB physical.



Source: Pure Storage.



Source: Pure Storage

Object metadata scales with the number of clients. To operate effectively, object storage must be able to handle many clients/threads/processes/hosts. We viewed an internal Pure S3 benchmark designed to GET 1KB objects from a FlashBlade configured with 15 x 17TB drives. It showed that, when adding threads and clients to S3 workloads, FlashBlade increased performance nearly linearly until it hit the client application limit. Its ability to read many threads in parallel makes this performance possible.

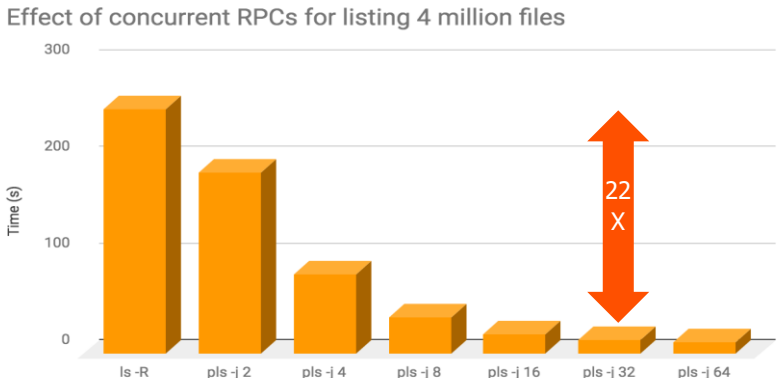
Also, according to Pure internal testing, FlashBlade performed 10x faster with 4KB PUTs on-premises than a leading cloud provider executing PUTS in the cloud. While ESG did not validate this methodology or the results, they

suggest the possibility of increasing performance using on-premises FlashBlade for object workloads that customers typically do in the cloud.

High File Count and Concurrency

As the number of threads increases, FlashBlade delivers higher IOPS, which results in faster metadata operations, lists, finds, and deletes. We viewed customer results showing file metadata performance scaling to 1.2M IOPS as clients were added.

Pure created a version of ls for Linux to demonstrate the increase in speed of listing files as threads were increased. We viewed a chart showing the decrease in time to list 4 million files as threads were increased, with 32 threads delivering 22x the performance of a single thread. We also viewed a live demonstration of listing files while increasing the number of threads; both file count and items per second increased as the number of threads increased from one to 64.



Source: Pure Storage

Why This Matters

Before consolidating applications, organizations must be certain that combining workloads will not degrade performance or interrupt productivity. The goal is to gain the reduced costs, reduced complexity, and greater efficiency of consolidation without impaired performance from the additional workloads and multiplication of threads/hosts/processes.

ESG validated the ability of FlashBlade to simultaneously support NFS, SMB, and S3 workloads with only minor interruptions during blade evacuation and addition, maintaining parity protection. We also viewed results demonstrating how FlashBlade systems in production are supporting billions of objects and are increasing file performance as threads increase.

The Bigger Truth

According to ESG research, 75% of respondents report that IT complexity has increased in the past two years, up from 64% in 2020. It is no surprise, then, that in the same research, respondents most often cited becoming more operationally efficient when asked to name their top digital transformation initiatives.[4] Consolidating workloads can significantly reduce complexity and increase efficiency, but only if performance and availability are maintained.

To this end, Pure Storage recently introduced FlashBlade as the Unified Fast File and Object storage platform. This scale-out, all-flash storage system provides unified, native file and object storage that is simple to manage and delivers high performance. FlashBlade's simplicity and multi-dimensional performance enable customers to confidently consolidate NFS, SMB, and S3 workloads on the same platform.

ESG validated:

- The simplicity that FlashBlade offers, including automated management and software-defined networking.
- The ease of creating file systems, object storage buckets, and remote replication.
- The ability to run both file and object workloads on the same system and monitor performance of all protocols from a single screen.
- The ability of FlashBlade to maintain performance for simultaneous workloads with both files and objects while blades were evacuated and reinstalled.
- Support for high object and file counts and increased performance with concurrent threads.

Pure believes that customers shouldn't have to worry about segregating workloads; instead, they should be able to run any application on any cluster. Consolidating on FlashBlade means that customers can share all applications on the same infrastructure, because it's not optimized for any specific I/O profile. It means fewer components to manage, less stranded capacity, better efficiency, and lower costs for power, cooling, and floorspace. It means that customers can use a single set of data for multiple use cases (such as analytics, AI/ML, IoT, etc.) instead of having to duplicate and triplicate data on different application silos.

ESG believes that most future infrastructures will include hybrid and multi-cloud support, so having fast on-premises object storage can put you ahead of the game. Pure's Unified Fast File and Object platform delivers the ease of use and multi-dimensional performance that lets customers confidently consolidate all their unstructured data workloads.

Of course, it is important to plan and test in your own environment to be sure that UFFO storage is right for your organization. But if you are looking to consolidate file and object storage with simple management and high performance, ESG recommends you take a good look at Pure FlashBlade.

[4] Source: ESG Research Report, *2021 Technology Spending Intentions Survey*, January 2021.

Enterprise Strategy Group is an IT analyst, research, validation, and strategy firm that provides market intelligence and actionable insight to the global IT community.

www.esg-global.com

contact@esg-global.com

508.482.0188