A Principled Technologies report: Hands-on testing. Real-world results.



DataCore vFilO lets users locate files across multiple NAS systems with a single search

In our testing, the solution also automatically moved files to better utilize storage resources

Network-attached storage (NAS) can provide straightforward file storage for an organization. Most users are comfortable performing basic file tasks navigating across a small number of folder and file hierarchies, known as file shares, on one NAS system. IT administrators are also comfortable with NAS technologies, but certain pitfalls tend to arise in environments that use this kind of storage.

When an organization outgrows their NAS capacity, they can either replace it or keep it in service and add a new system. Upgrading to a new system can improve performance, security, and data privacy, but can be expensive and require end users to adjust to a new set of folder and file hierarchies. For these reasons, many organizations accumulate a variety of NAS systems over the years.

With file shares across disparate NAS systems, administrators must monitor capacity on each of them. If some systems are almost full while others have plenty of space, it's tempting to move files, but doing so can break links to files that end users have embedded in their documents. Maintaining multiple NAS systems also requires users to search one hierarchy after another when they are trying to locate a file.

DataCore[™] vFilO[™], a software solution that virtualizes file and object storage, can help with these challenges. We tested it and found that it let users access files the same way they always have and allowed them to conduct a single search to locate files across multiple NAS systems. It also automatically moved files to make the best use of existing storage capacity, which help organizations delay the purchase of new hardware.

A global namespace

DataCore vFilO takes the file shares it controls across an organization's NAS systems and creates a unified way to view all of them, known as a namespace. This lets users locate a file with a single search rather than having to search each file share individually. Behind the scenes, vFilO migrates files from one volume to another to make efficient use of storage capacity, but the file and folder hierarchy remains intact. Document links to file shares work consistently even when the physical locations of the files change.

DataCore vFilO provides a full set of data placement policies intended to let admins set the availability, performance, cost, and location preferences for users' data while preserving the file and folder hierarchy on which users depend.

DataCore vFilO can also move inactive data to low-cost object storage automatically by policy, freeing space and load on existing NAS systems, and possibly delaying the need to purchase new NAS systems.

An overview of our testing

Once we set up the DataCore vFilO environment, we tested its ability to automatically move files between volumes without changing the way that file structures appear to users. We also tested its ability to let users locate files across multiple NAS systems with a single search.

Our test environment included one NAS system and two file servers, which we filled to varying degrees with files of a variety of types and sizes, and three mid-range dual-socket servers, on which we installed the DataCore vFilO software.

We also included a cloud volume in our testing to show the cloud offload capabilities of vFilO. We created an object bucket in Amazon Web Services (AWS) S3, then provided the access credentials for this bucket in the vFilO console.

We present the complete configuration details of our servers and NAS systems in the science behind this report.

How vFilO works: Assimilation

After setting up our file servers and installing the DataCore vFilO software on three additional servers, we "assimilated" the file shares from the three file servers into the DataCore vFilO system. DataCore uses the concept of assimilation to describe the process of traversing the tree of each file share and gathering metadata about every file. DataCore vFilO can assimilate a variety of types of NAS systems, including several commonly used filers, NFS on Linux or Windows, and SMB on Windows. This assimilation process creates a new globally searchable namespace, which can be read-only or read-write, and can run all vFilO services. DataCore vFilO configuration on this process in the "Assimilation" chapter of the DataCore vFilO Configuration Guide.

What we learned in our testing

Testing the vFilO Data Mobility features

DataCore vFilO Data Mobility moves files between volumes based on rules—mobility percentages and objectives—that the IT administrator sets up.

Say the administrator wants Volume A to always keep 25 percent of its capacity open. They would then set the mobility percentage for that volume to 75 percent. Before the used space on Volume A exceeds that

percentage, vFilO automatically seeks other volumes whose current usage is below their mobility percentage, and migrates some of the files from Share A to the other volumes.

vFilO has a rich set of objectives that administrators can apply to entire shares and individual files. Examples include "exclude from object-volumes," "place on object-volumes," "optimize for capacity," "place on <volume-name>," "place in <volume-group>," "availability-5-nines," and several others. Administrators can also create custom rules based on the activity level of the file or folder, the time elapsed since last access of the file or folder, and many other conditions.

We applied some data placement objectives so that vFilO would automatically migrate files to achieve a more optimal balance of capacity and performance while maintaining the original file and folder hierarchy. We set mobility percentages ranging from 75 percent to 95 percent. We also specified that some files and folders could migrate to the cloud.

In Figure 1, we zoom in on the Volume list screen at two points, before vFilO moved the files in response to the mobility settings and objectives we put in place and after. The purple arrows show how capacity changed from "before" to "after." Note that after we assimilated the original volumes, we created two new volumes to provide additional capacity for storage. These two new volumes appear at the bottom of the "after" screenshot.

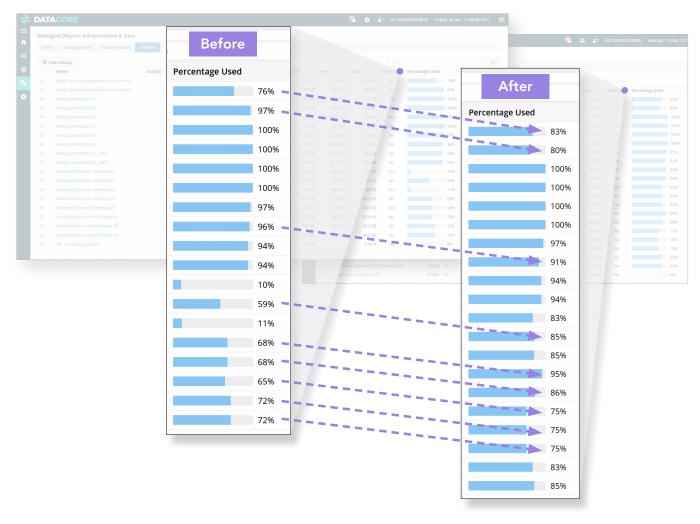


Figure 1: Local volume usage before and after DataCore vFilO automatically moved files in response to mobility percentages and data placement objectives. Source: Principled Technologies

In Figure 2, we again zoom in on the Volume list screen before and after vFiIO moved the files in response to our mobility settings and objectives, this time focusing on usage of the cloud volume. In the "before" screenshot, the cloud volume contained no files, while in the "after" screenshot, vFiIO has moved 329 GB of files to the cloud volume.

	Volume Group IGPS af IGPS af IGPS af IGPS af IGPS af IGPS af IGPS af	Storage System data01.ptech.com data01.ptech.com NetApp NetApp NetApp	Used 455 G8 583 G8 1.62 T8 1.63 T8 1.65 T8	Free Tot 144 G8 165 G8 3.13 G8 7.43 G8	599 GB 599 GB 1.62 TB	Q. Search Read Only Perc No No No No	entage Used 76% 97% 100%	Actions		Used F	Tree			
	IOPS all IOPS all		455 GB 583 GB 1.62 TB 1.63 TB	144 GB 16.6 GB 3.13 G8	599 GB 599 GB 1.62 TB	No No	76%	© / 1 © / 1		Used F	ree			
	IOPS all IOPS all IOPS all IOPS all IOPS all		583 GB 1.62 TB 1.63 TB		599 GB 1.62 TB		97%	@ / T		Used F	ree			
	IOPS all IOPS all IOPS all IOPS all IOPS all		1.62 TB 1.63 TB		1.62 TB					Used F	ree			
	IOPS all IOPS all IOPS all						100%							rcentage Used
										497 GB				
	IOPS all		1.65 TB				100%	۰ / ۱		479 GB	120 GB			
					1.66 TB		100%	۰ ۲		1.62 TB				
			1.52 TB		1.52 TB		100%	• / I		1.63 TB	7.42 GB			
						No	97%	@ / I		1.65 TB	5.51 GB			
			2.15 TB	88.5 G8	2.24 TB		96%	• / 1		1.52 TB	694 MB			
	IOPS all		644 GB	41.3 GB	685 GB		94%	@ / I						
			524 GB		555 GB		94%	@ / I		2.18 TB	223 GB			
			103 GB	897 GB	1.00 TB		10%	• / I		644 GB	41.3 GB	685 GB	No	_
pt-grotius01			0 IOPS all,	object-volu	imos	Cloud				0.00 B	1.1 GB	Re	efore	
			olors all,	object-voit	ames	Ciouu				0.00 D	154 GB	Бе		
										851 GB	149 GB	1.00 TB	Yes	_
							68%	• / 1		570 GL				
										516 GB				
										736 GB	247 GB			
										738 GB	246 GB			
											247 GB			
										834 GB	166 GB			
										851 GB	1 GB			
				o 🔵 🛥 Cloud:						329 GB	1			
	-03 0 -02 0	01 0.025 at 02 0.025 at 03 0.025 at 040 0.025 at	03 0 ICPS all LinuxHieserVer	0.0 0.0P5 al Linearingterver 708.GB 0.0 0.0P5 at UnudfileServer 712.06	0.0 0.055 all Lunstriedenter 708 68 276 68 0 00595 all Lunstriedenter 712 08 271 08 0 00595 all Lunstriedenter 712 08 271 08 0 X0595 all object volumes Choud 0.059 — —	6.03 0 C/OFS all Lowertescenser 708 08 276 68 563 683 <	L01 0.075 L1 LundrinstantWit 708 08 276 08 983 08 No 60 0.075 all Unstitivitiener 712 08 271 08 983 08 No 0.0015 all object volumes Cloud 0.001 — 0.001 No	0.0 0.05% 0.1 Loweringserver? 706.08 226.63 963.63 No 77% 00 000% at Unutifiedence 712.06 271.68 963.06 No 77% 0100% at, object votumes Cloud 0.06 - 0.008 No 75% 0100% at, object votumes Cloud 0.06 - 0.008 No 96	0.01 0.05% 0.0 Linuxindentif 708.68 926.68 963.68 No 726 @ / I 00 0.05% dl. Unutlikener 712.06 271.03 983.06 No 728 @ / I 0.005 el, object valumes Citud 0.005 – 0.005 No 06 @ / I 0 Wordswitheleveronth direkters/citud 0.005 – 0.005 all Windswitheleveronth direkters/citud 00.05 all Windswith level	0.01 0.05% 0.01 0.04 0.06 0.05 0.01 0.07 0.07 0.07 0.07 0.01 <t< td=""><td>Cell D CPS all Contribution C 41 05 24 05 24 000 1000 60 60% 60% 10000 10000 10000 10000 10000 10000 <th1< td=""><td>col 0.05% col 0.004 0.01 0.005 0 <th0< th=""> <th0< th=""></th0<></th0<></td><td>Old D/D5* al Loundration Call 0 Pail 00 Pail 0</td><td>col 0.055 cit Lower body Col 90.06</td></th1<></td></t<>	Cell D CPS all Contribution C 41 05 24 05 24 000 1000 60 60% 60% 10000 10000 10000 10000 10000 10000 <th1< td=""><td>col 0.05% col 0.004 0.01 0.005 0 <th0< th=""> <th0< th=""></th0<></th0<></td><td>Old D/D5* al Loundration Call 0 Pail 00 Pail 0</td><td>col 0.055 cit Lower body Col 90.06</td></th1<>	col 0.05% col 0.004 0.01 0.005 0 <th0< th=""> <th0< th=""></th0<></th0<>	Old D/D5* al Loundration Call 0 Pail 00 Pail 0	col 0.055 cit Lower body Col 90.06

Figure 2: Cloud volume usage before and after DataCore vFilO automatically moved files in response to mobility percentages and data placement objectives. Source: Principled Technologies

When vFilO moves a file to a new location, you can find its back-end storage volume by looking at the Storage volume column. Clicking the specific storage volume provides details about the resource (see Figure 3).

DATAC	ORE						LT DC14	ID44D2DCEB1D	Friday, 8 Ma	ay, 11:11 A	MEDT				
			Files Capacit	ty Perf	formance	Alignmer	nt								
Shares	main NS08	Performance									C				
Туре	Files / Directories	Size	Applied Objectives	Activ	ve Objectives	Storage \	olume	Access Time	Alig	gnment	Actions				
Directory	.snapshot	-	12 Objectives 🧖	6 Ob	jectives	_		5/8/2020 11:	11 AM						
Directory	.collections	-	-	-		_		5/8/2020 11:	11 AM						
File	vdbenchdatafile	524 MB	12 Objectives 🦨	10 0	bjectives	NetApp	_0001	4/17/2020 2:*	14 PM		0 (†				
Vo	olume Properties			and the second s							×				
										💉 Ed	lit				
	Volume Name: NetAp	op::svm:Data_08000	D		Storage Nam	ne: NetApp									
	IP Address: 10.211.33	3.125			Storage Type	e: NetApp C	-Mode								
	Availability: Available				Storage IP Ad	ddress: 10.	211.33.120								
	Operational Targets														
			80												
	Capacity			% 100000	80% X N	Aobility Thr	eshold								
	Read Latency				100000 M	S									
	,			MS 100000											
	Write Latency			MS	100000 M	S									
				1110							_				
	Capabilities														
	Performance	Defined		F	Protection	De	fined								
	Read IOPS	5055 OPS		A	Availability	99.	9% (3-9s)								
	Write IOPS	4703 OPS		C	Durability	99.	99% (4-9s)								
	Read Latency	1.484534292 MS													
1//////	Write Latency	2.023653627 MS													
	Read Bandwidth	343.32 MBPS													

Figure 3: DataCore vFilO Volume Properties screen. Source: Principled Technologies

After completing the assimilation process and some data migration, we can view the storage systems' information from the vFilO console, which shows the name and type of each storage system as well as its capacity information (see Figure 4).

	🖗 🖘 https://10.21	τ	nodes								⊠ ☆	lii\	ŧ (3 🐠
DATACOR	E							9	≜ ⊁•	DC14D44I	02DCEB1D	Thursday, 30 Apr, 9	5:53 PM	EDT
Managed Objec	cts: Infrastructure & Da	ta												
Shares Storag	ge Systems Volume Groups	s Volumes												
Add Storage S	ystem									Q Search			×	C
Name		Volumes	Vol	lume	Statu	s	Туре	Used	Free	Total	Percentage	e Used	Actio	ns
data01.ptnet.princ	ipledtech.com	2	2	0	0	0	DataCore	836 GB	362 GB	1.20 TB		70%	۲	0
WindowsFileServe	r	7	0	0	0	0	Windows File Server	2.93 TB	3.28 TB	6.20 TB		47%	۲	0
NetApp		8	8	0	0	0	NetApp C-Mode	11.6 TB	103 GB	11.7 TB		99%	۲	0
LinuxFileServer	+ Volume	4	8	0	0	0	Other	1.97 TB	980 GB	2.95 TB		67%	۲	0
Cloud		1	0	0	0	0	AWS S3	559 GB	0.00 B	0.00 B		0%	۲	0
dsx-a01.dm01.loca	d	0	0	0	0	0	DataCore	0.00 B	0.00 B	0.00 B		0%	۲	Ø
dsx-a02.dm01.loca	d	0	0	0	0	0	DataCore	0.00 B	0.00 B	0.00 B		0%	۲	0
dsx-a01.ptnet.prin	cipledtech.com	0	0	0	0	0	DataCore	0.00 B	0.00 B	0.00 B		0%	۲	0

Figure 4: vFilO view of the storage systems. Source: Principled Technologies

Testing moving files to the cloud

To enable automatic data migration to the cloud, we created a custom "Objective" that would migrate all MP4 files older than one year to the cloud (see Figure 5). Any MP4 file newer than one year would follow the default objectives. This was a relatively simple custom objective, and we could have added more criteria to this objective.

4	DataCore vFilO	× +									-	o x
€→) C' 🛈	🖸 🔒 htt	ps:// 10.211.211	.1/#/thresholds/objectives/create/step3				♡☆		⊻ ∥\ 🗉 🦉	۲	Ti
4	DATACORE				20	È	7.	DC14D44D2DCEB1D	Wedne	sday, 20 May, 1:20 PN	/I EDT	
■ ∩	Edit and Update	Existing Ob	jective	1 Name	2 Define Rules	3 Rev	view					
*	Name	A	Rules									
0	Move old MP4s to c	loud	IF	File Name: *.mp4 AND Apply to Live Files	1 year inactive - 1, 7 years inactive - 1			PLA	CE ON	default-objective place-on-Cloud		\boxtimes
	Description	1										
¢	Move all MP4 files r one year old to the									Cancel Save	e Object	ive-

Figure 5: Creating a custom objective. Source: Principled Technologies

Client view

After we assimilated all the file servers into vFilO, we mounted the "main" vFilO file share that we created previously onto our application server. Once we did so, all of the file shares from all of the file servers were visible and accessible (see Figure 6). We mounted this "main" share using NFS.

🛫 💆 📜 = м	anage main (\\10.211.211.4) (\	/:)			- 🗆 ×		
File Home Share View Driv	ve Tools				~ 🤇		
Pin to Quick Copy path access	Move Copy to* to*	New item • The basy access • New folder	Properties	Select all Select none Invert selection	,		
Clipboard	Organize	New	Open	Select			
\leftarrow \rightarrow \checkmark \uparrow \checkmark \Rightarrow This PC \Rightarrow main (\\	10.211.211.4) (V:) 🔉			✓ ບ Search	main (\\10.211.211.4) (🔎		
> 📌 Quick access	Name		Date modified	Size			
	assimilation-logs		4/23/2020 1:14 PM	File folder			
🗸 🍃 This PC	📕 fileshare01		4/24/2020 11:27 AM	1 File folder			
🔉 🧊 3D Objects	Fileshare02		4/23/2020 12:55 PM	File folder			
> 📃 Desktop	📙 linuxdata01		4/23/2020 1:07 PM	File folder			
> 🗎 Documents	📜 Linuxdata02		4/23/2020 1:36 PM	File folder			
> 🚽 Downloads	📙 Linuxdata03		4/23/2020 1:36 PM	File folder			
> Music	NS01		4/17/2020 2:00 PM	File folder			
 Pictures 	NS02		4/17/2020 2:04 PM	File folder			
	NS03		4/17/2020 2:07 PM	File folder			
> 📑 Videos	NS04		4/17/2020 2:08 PM	File folder			
🔉 🐛 Boot drive (C:)	NS05		4/17/2020 2:09 PM	File folder			
🔉 🥧 Data-SSD (D:)	NS06		4/17/2020 2:10 PM	File folder			
> 🕪 main (\\10.211.211.4) (V:)	📕 NS07		4/17/2020 2:12 PM	File folder			
	NS08		4/17/2020 2:14 PM	File folder			
> 学 Network	Performance		4/30/2020 11:48 AM	File folder			

Figure 6: Client view of all the folders presented by vFilO. Source: Principled Technologies

Search once rather than repeatedly

One of the benefits of a global namespace is that all the previously scattered file shares are visible to end users as though they were in one file structure. As a result, one can issue a single search command for a filename and vFilO will search the entire namespace of all the file shares for the desired file. Without vFilO, a user would have to conduct a separate search for each of the original file shares. Furthermore, once they found a file, they would not know whether it was the most recent version. As part of our testing, we successfully validated that vFilO global namespace search was fully functional.



Conclusion

Organizations that use multiple NAS systems for file storage face a number of inconveniences that DataCore vFilO can minimize. By providing a single, global namespace, vFilO makes it easier for users to find the files they are looking for. IT administrators can also benefit from vFilO policies that automatically migrate files to the appropriate local or cloud storage and consolidate multiple older file servers, and the company can potentially reduce costs by avoiding or delaying the purchase of new storage hardware. In our tests, vFilO moved files and made them accessible to users in the way its documentation claims.

Read the science behind this report at http://facts.pt/nagxt0w ▶





Principled Technologies is a registered trademark of Principled Technologies, Inc. All other product names are the trademarks of their respective owners. For additional information, review the science behind this report.

This project was commissioned by DataCore.