

Case Study:

How a Global Pharma Company Used cunoFS to Access its Private Object Storage System

One of the top five global pharma companies has extensively tested cunoFS with its on-prem object storage system to give POSIX access directly to unmodified objects in buckets.

cunoFS is client-side software that gives applications high-performance, high-throughput POSIX-compliant file access to unmodified data in object storage like Amazon S3 or on-premises or cloud alternatives. This is achieved without modifying the objects.

Performance is a major differentiator for cunoFS, in addition to an intuitive developer experience and file access at scale. The product's performance exceeds 50 Gbps read/write access directly to native objects as if they were a part of the user's filesystem, on a single node. cunoFS has been benchmarked at more than 10 Tbps aggregate throughput using 256 nodes.

In this case study, we cover how one of the top five global pharma companies achieved a 5–13x speedup in operations on both large and small files with cunoFS compared to its S3-compatible on-prem storage system.

File access at large pharma companies: a complex use case

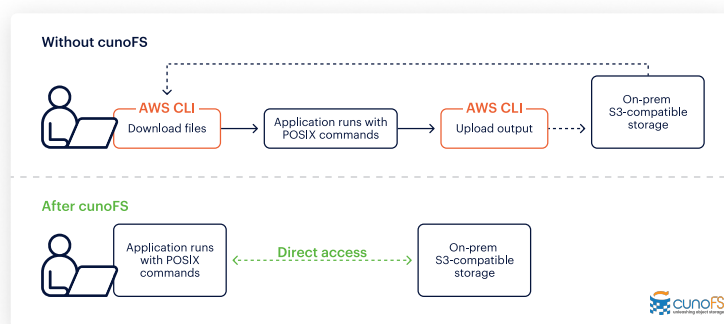
Before encountering cunoFS, the pharma company's employees did not previously have POSIX access to the data in the on-prem object storage system (ObjStore). In order to use data from the ObjStore with POSIX commands or workflows, they had to stage the data into an on-prem high-performance filesystem; and in order to manage the data they had to use special object storage commands (AWS CLI S3 commands).

This workflow was confusing to users. The pharma company's team wished there were a way to access files using standard POSIX commands and workflows directly on the unmodified object data in the ObjStore, without needing to stage the data and without needing special commands.

In addition, staging the source data and migrating the resulting data was slow, frequently adding hours to researchers' workflows.

Speeding up data access would directly result in increased productivity for key highly qualified technical staff.

Top-5 global pharma company file access workflow



Evaluating cunoFS to help with developer experience and productivity

In order to evaluate cunoFS, the pharma customer performed multiple tests using various file operations. The performance of cunoFS-enabled POSIX commands to access the on-premises S3-compatible data store was compared against AWS CLI S3 commands. All testing was done with default options, without any tuning. The customer identified some of their real-life workloads as involving large files, while others involved large numbers of small files, so performance was tested for a very large number of small files, a large number of medium-sized files, and one very large file.

File sizes used during testing

Small files	1,000,000 files x 1 MB each
Medium-sized files	1,000 files x 1 GB each
Large files	1 file x 1 TB

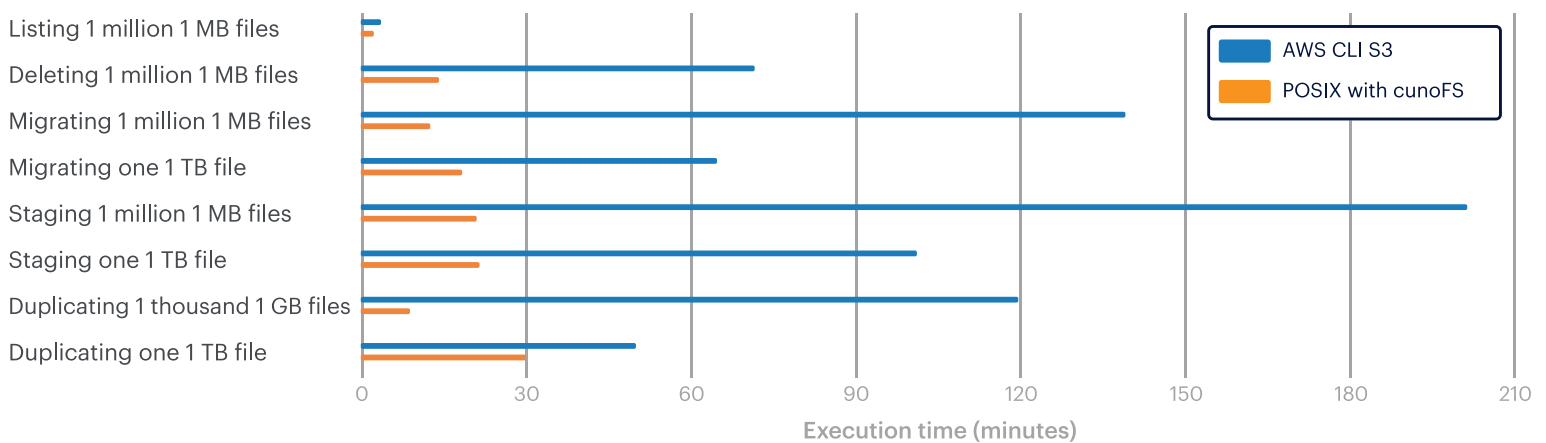
Functionality testing of a variety of common POSIX commands

One of the key requirements for the pharma company was for all POSIX commands to work correctly.

Upon loading cunoFS, the company's objects in the unstructured object storage were represented by virtual files in the filesystem, and buckets in the object storage were represented by virtual directories.

By executing the POSIX commands on the virtual files and virtual directories, the pharma customer verified that all of the commands functioned correctly on objects using cunoFS.

Summary of speed-up achieved using cunoFS



File management: stress-testing POSIX commands to list and delete objects with large files and many small files

The next step was to test the performance of list and delete commands with cunoFS.

For listing very many small files, POSIX with cunoFS was **twice as fast** as the AWS CLI S3 command (saving more than 100 seconds of time per query), and it was **13x faster** than AWS CLI for a large file.

For deleting very many small files, POSIX with cunoFS was **5x faster** than the AWS CLI S3 command (reducing the deletion time from 72 minutes to 14 minutes), and it was **19x faster** than AWS CLI for a large file.

Migrating data from an on-prem POSIX-compliant filesystem to an on-prem object storage solution

The pharma company had a frequent operational requirement to migrate numerous files and large amounts of data into the object storage, which is their main data store. This utilized heavy amounts of time and resources. The performance of cunoFS using fpsync (rsync) was significantly superior in all cases.

For migrating very many small files, POSIX with cunoFS was **11x faster** than the AWS CLI S3 command (reducing the migration time from 2.3 hours to 12 minutes), and it was **3.6x faster** than AWS CLI for a large file (reducing the migration time from 1 hour to 18 minutes).

Staging data from an on-prem object storage solution to an on-prem POSIX-compliant filesystem and vice versa

While staging of data could be avoided using cunoFS (since POSIX commands could run directly on the objects using the virtual files provided by cunoFS),

the company was still interested in staging time, due to varying user preference or for the period after adoption of cunoFS before workflows were adjusted to use direct POSIX access.

The performance of cunoFS was significantly superior in all cases, with the comparative results similar for staging in each of the two directions.

For staging (copying) one million 1 MB files, POSIX with cunoFS was **9x faster** than the AWS CLI S3 command (reducing the staging time from 3.3 hours to 20 minutes), and for a large file it was **4x faster** than AWS CLI (reducing the staging time from 1.7 hours to 20 minutes).

Duplicating objects between two buckets on an on-prem object storage solution

It is sometimes necessary to duplicate objects to run operations on the copies.

For duplicating many medium-sized objects, POSIX with cunoFS was **13x faster** than the AWS CLI S3 command (reducing the duplication time from 2 hours to 9 minutes), and for a large file it was 65% faster than AWS CLI (reducing the duplication time from 50 minutes to 30 minutes).

Running a demanding Linux application workload directly on cunoFS

The company ran an example workload with the data being accessed directly from the buckets with cunoFS, instead of running the same workload on local files that were downloaded from object storage and placed on the local filesystem.



Running POSIX workloads directly on objects in buckets would avoid and eliminate staging time altogether. In addition, **the actual execution time using cunoFS was even faster**, compared with running the workload both (i) on a high-performance filesystem on local files, and (ii) on the pharma customer's in-house genomics platform on local files.

Building a business case for more efficient object storage access

While the numbers were interesting, the pharma company's technology team needed to tie the performance improvements to business benefit. For the pharma company, the benefits came in four categories.

Driving object storage adoption

Most organizations buy or use object storage solutions due to their scalability and cost, whether on-prem or in the cloud. However, they frequently face the difficulty of lack of adoption by their users, mainly due to how different the user experience is compared to standard file access.

Without cunoFS, the lack of POSIX access, the inconvenience of managing object data, the need to stage data prior to running workflows, and the slow migration and duplication speeds all feel discouraging to the end users, resulting in lower adoption of object storage at the organization.

cunoFS speeds up the adoption by providing high-speed transparent POSIX access, eliminating the need for staging of data, and providing 5–13x faster speed of migration and data management. cunoFS thereby drives frictionless adoption at scale by users of object storage solutions, enabling the organization to achieve the economic benefits that it expects from deploying object storage solutions.

Reducing the need for expensive scratch storage

In traditional object storage setups, a user needs to stage files from object storage into a faster, POSIX local filesystem before running their application. This local storage is often more limited and much more expensive.

Using cunoFS removes the need for staging files, significantly reducing the need for expensive scratch storage and reducing the size requirements for high-performance filesystems. This is a direct cost reduction to the organization.

Reducing waiting time, even with legacy workflows

Even for workflows and practices that are already performed with object storage, use of cunoFS saves a significant amount of waiting time for users, thereby improving organizational productivity.

cunoFS eliminates the time needed to stage data into storage attached to compute before workflows can start, which is a significant amount of time saved. In the case where user preference or legacy workflows mean that staging is still performed, cunoFS reduces this staging time from hours to minutes.

cunoFS also reduces the time to migrate new data into object storage from hours to minutes, and is also more efficient at manipulating the data after it has been migrated to the object storage.

Offering POSIX access to deliver a better user experience

The convenience and comfort that users experience from cunoFS's provision of familiar POSIX access to and management of their data has direct benefits for the organization's staff.

By eliminating friction and providing an interface that users like and are familiar with, the organization can fully adopt object storage solutions while getting faster user buy-in and maintaining productivity.

A detailed performance comparison of cunoFS vs. existing solutions

1. Functionality tests were performed demonstrating that cunoFS allows execution of the following common POSIX commands on the contents of a bucket in the on-prem object storage solution (all tests were successful):

ls [-ltrha]	du [-h]	df [-h]	rm [-r]	mkdir	pwd
cat	rmdir	touch	locate	find	grep
head	tail	diff	tar	chmod	chown
zip	unzip	more	wc	cd	

2. Performance of the POSIX commands to list and delete objects with cunoFS enabled were compared with the equivalent AWS CLI S3 command:

Command	Execution time (seconds)		
	1,000,000 files x 1 MB	1,000 files x 1 GB	1 file x 1 TB
ls -lh	101	0.373	0.19
aws s3 ls --human-readable --summarize	204	0.863	2.42
Relative speed of ls with cunoFS, vs. aws s3 ls	202%	231%	1,274%
rm -fr	858	11.8	0.153
aws s3 rm --recursive	4,308	64.8	2.880
Relative speed of rm with cunoFS, vs. aws s3 rm	502%	549%	1,882%

3. Times to migrate data from an on-prem POSIX-compliant filesystem to an on-prem object storage solution were compared:

Command (rsync equivalents)	Migration time (seconds)		
	1,000,000 files x 1 MB	1,000 files x 1 GB	1 file x 1 TB
<code>fpsync -n 16 -o "--inplace"</code>	755	356	1,089
<code>aws s3 sync</code>	8,358	3,764	3,903
Relative speed of <code>fpsync (rsync)</code> with <code>cunoFS</code> , vs. <code>aws s3 sync</code>	1,107%	1,057%	358%

4. Times to bidirectionally stage data between an on-prem object storage solution (ObjStore) and an on-prem high-performance filesystem (HPFS) were compared:

Command	Staging time (seconds); from_ObjStore_to_HPFS / from_HPFS_to_ObjStore		
	1,000,000 files x 1 MB	1,000 files x 1 GB	1 file x 1 TB
<code>cp -r</code>	1,265 / 1,027	208 / 388	1,293 / 925
<code>aws cp --recursive</code>	12,059 / 8,377	3,061 / 3,757	6,075 / 3,866
Relative speed of <code>cp</code> with <code>cunoFS</code> , vs. <code>aws cp</code>	953% / 816%	1,472% / 969%	470% / 418%

5. Times to copy data between two buckets on an on-prem object storage solution were compared:

Command	Copy time (seconds)	
	1,000 files x 1 GB	1 file x 1 TB
<code>cp -r</code>	548	1,826
<code>aws cp --recursive</code>	7,190	3,016
Relative speed of <code>cp</code> with <code>cunoFS</code> , vs. <code>aws cp</code>	1,312%	165%

6. Times to execute a genomic BWA-MEM POSIX workload were compared:

Location of data / Location of compute	BWA-MEM execution time (seconds)
Object storage / on-prem HPC compute node (ObjStore)	582
High-perf filesystem / High-perf filesystem (HPFS)	601
Relative speed on ObjStore with <code>cunoFS</code> , vs. HPFS	103%
In-house genomics platform / In-house genomics platform (GP)	723
Relative speed on ObjStore with <code>cunoFS</code> , vs. GP	124%

Next steps for the pharma company

The pharma company identified six use cases for `cunoFS` at the beginning of the proof-of-concept project:

1. To give their workloads convenient and fast POSIX access to their on-prem private object storage system, without altering the objects.
2. To burst to the cloud in an efficient and seamless way when their in-house cluster compute is overloaded and the necessary files are in their private object storage system.
3. To reverse-burst from the cloud back to their in-house compute cluster when the cluster compute is available and the necessary files are in a cloud-based object storage system.
4. To stage data from either their on-prem high-performance filesystem or their on-prem object storage solution to a cloud-based S3 storage system or other cloud-based system.
5. To transparently and quickly move daily snapshots (backups) of their cluster to their private object storage system.
6. To augment the capabilities of their in-house genomics platform to provide its users with transparent POSIX access to S3 objects.

While in this PoC they focused on task 1, they are still interested and are looking forward to exploring use cases 2–6 and measuring the relevant performance impact.

`cunoFS` proved to be a valuable solution for the pharmaceutical company's use case, enabling efficient and convenient POSIX access to their on-prem object storage while delivering superior performance and numerous business benefits.

Interested in seeing if `cunoFS` would be a fit for your use case?

Get a free trial on our website cuno.io



Come find us at: ideaSpace,
Hauser Forum, 3 Charles Babbage Road,
Cambridge, CB3 0GT, UK



Call us on: +44 (0)1223 655651



Email us: info@cuno.io

