

## Overview

**cuno** enables native, unmodified and accelerated access to lower cost, infinitely scalable object storage buckets reducing the need for more expensive storage tiers and without introducing bottlenecked file gateways.

## Unmodified Objects

Files are stored as unmodified objects. No gateways, no need for servers. Objects remain fully accessible to other systems with native object storage access.

## Transparent Usage

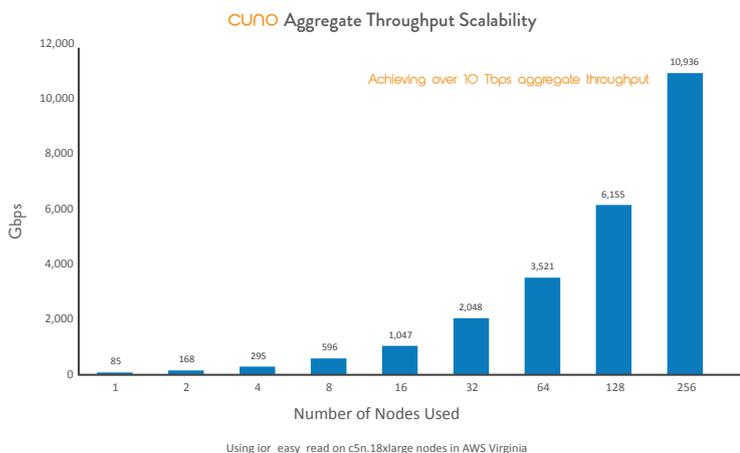
Cloud-enable and random-access stream data directly to your existing Linux apps, tools, POSIX commands, pipelines, workloads and containers. No data needs to be written to disk or pre-downloaded.

## High-Performance

**cuno** provides much faster transfer of data to and from object storage. Benchmarked at 50+ gigabits/sec per node on AWS, and at 10.9 Tbps aggregate across nodes. That's faster than most high performance file-systems.

## Significant Cost Reductions

**cuno** enables organisations to quickly and easily broaden the usage of object storage without the need for code changes or updates to existing mission critical applications.



## Features



### Unmodified Objects

Files stored as objects, accessible to Object-Native applications. Objects as files.



### POSIX Compliance

Run complex Linux workloads with ease with POSIX compatibility.



### Accelerated Access

Ultra fast transfer of data to/from Object Storage up to 50+ gigabits/sec per node.



### Hybrid & Multi-Cloud

Access on-premise or cloud object storage to use or transfer data as appropriate.



### Easy to Use

No admin privileges required to install the client to natively read/write object storage.



### Permissions Control

With POSIX ACL support. Changes in access can even be mirrored back to the native-object interface (S3 only).

**cuno** transparently bridges the gap between workflows and object storage



For more information, please contact us:  
[info@cuno.io](mailto:info@cuno.io)  
[www.cuno.io](http://www.cuno.io)

URI-based access

```

~ cuno
(cuno) ~ cd s3://bkt
(cuno) s3://bkt tar xfb az://pg/bkt/archive.tar.gz
(cuno) s3://bkt chmod a+x script.sh
(cuno) s3://bkt grep -R test > tests.txt
    
```

Path-based access

```

~ cuno
(cuno) ~ cd /cuno/s3/bkt
(cuno) /cuno/s3/bkt tar xfb /cuno/az/pg/bkt/archive.tar.gz
(cuno) /cuno/s3/bkt chmod a+x script.sh
(cuno) /cuno/s3/bkt grep -R test > tests.txt
    
```

### Why cuno?

By far the dominant majority of applications, including scientific ones, only deal with file-based storage. Even applications that talk object, usually talk with just one API, such as S3, but not Azure. There are solutions that try to bridge the gap between file-based applications and object, but they suffer from various problems such as POSIX incompatibility, severe performance bottlenecks, vendor lock-in, files stored as scrambled objects, and more.

**cuno** addresses all these issues while providing incredible performance and scalability. **cuno** does not deploy a gateway or need servers. It doesn't scramble or modify content, which means files stored with **cuno** are directly accessible to object-native applications, and visa-versa. Performance scales with instances, up to the limits of the object storage itself. It doesn't require admin access, a kernel module, or a FUSE mount, and can be easily injected into Docker and other containerised environments. Even applications that natively support object storage run up to 10x faster with **cuno** instead.

