ondat™

architectingit

Report

Performance Benchmarking Cloud Native Storage Solutions for Kubernetes

Category: Technical Evaluation

Release 1.0.1 | 3 March 2021

Contents

Performance Benchmarking Cloud Native Storage Solutions for Kubernetes – First Edition

Published by Brookend Limited. Document reference number BRKSW0132.

No guarantees or warranties are provided regarding the accuracy, reliability or usability of any information contained within this document and readers are recommended to validate any statements or other representations made for validity.

Executive Summary

Modern storage media offers high throughput and low latency, with the capabilities of the latest NVMe SSDs delivering IOPS and bandwidth once expected from entire storage solutions only a decade ago.

Storage is evolving and one area of growth is the approach of software-defined storage solutions that look to deliver persistent storage for container-based application infrastructure. Like all storage solutions, there is disparity in the ability to exploit the benefits of the underlying media and deliver the best performance for lowest TCO (total cost of ownership). Some will achieve better results than others.

This paper documents a series of tests that aim to highlight the differences in performance between container attached storage (CAS) solutions with respect to standard I/O profiles. While CAS solutions provide storage that meets the needs of the application, the operational benefits of CAS have to be delivered without impacting performance from the perspective of the application. The performance of CAS solutions is perhaps even more important than SAN or HCI storage as containers offer such a lightweight and efficient way to run application code. Any performance overhead will naturally be more obvious and have greater impact with solutions that operate with such efficiency.

This report contains the results from testing across four of the leading CAS solutions on the market today. These are Ondat, Longhorn, Rook/Ceph and OpenEBS. The testing process uses free open-source software fio to generate a mix of random and sequential I/O across four scenarios: Local volume only, local volume with replica, remote volume, and remote volume with replica.

Testing was performed using a pre-determined hardware and software configuration that is consistent for each platform. The tests use NVMe SSDs for persistence, offering the ability to deliver high performance to a containerised application. These four tests demonstrate the capabilities of each software platform and how they exploit local resources such as system memory, while mitigating the overhead of networking and maintaining data protection.

These four tests demonstrate the capabilities of each software platform and how they exploit local resources such as system memory, while mitigating the overhead of networking and maintaining data protection.

The results show that Ondat performed better in all tests compared to the three competitors, with a strong set of results in local read performance with and without a mirrored replica. From the same set of hardware resources, Ondat delivered greater throughput, bandwidth and with lower I/O latency than all of the competitor solutions.

Introduction

Data storage is an essential part of every computing system, providing long-term persistence to application data. During the past 20 years, shared storage in the form of storage area networks (SANs) has dominated the industry. HCI (hyper-converged infrastructure) appeared as a competitor around ten years ago, distributing storage across multiple server nodes in a cluster of hardware.

With the rapid adoption of container-based applications and Kubernetes, the need for persistent storage for containers has evolved from SAN-based plugins to container-native solutions that deliver storage within the same or an adjacent cluster to an application.

In this report, we look at some of the most popular CAS solutions available today and benchmark them using standard storage performance tools. The aim is to demonstrate that I/O performance is equally important with container-based workloads, especially those running production applications.

Container attached storage

Container-based applications are now a key part of the infrastructure landscape. Containers, initially based on Docker and predominantly now through Kubernetes, offer lightweight, portable and efficient services deployment and a transition to microservices.

Although initially expected to be ephemeral, containers have become increasingly dependent on persistent storage. It has become clear that application-based data protection and services will not suit the majority of application requirements and introduces unnecessary replication and I/O overhead. As a result, applications expect locally connected persistent volumes that offer the features of traditional storage, namely data resiliency, performance and data services.

CAS has emerged as one option for mapping physical storage to container-based workloads. In these solutions, persistent storage is implemented as software-defined storage (SDS) running in containers on the same infrastructure as the applications. The components of the storage solution run as a set of resilient containers across nodes within a container-based cluster. CAS solutions offer a number of benefits to container-based applications:

- Storage devices are exposed locally by processes running on the same server or virtual machine as the application. There's no intervening network such as a SAN that can introduce latency (except for data protection overhead).
- CAS resiliency and performance can scale dynamically with a container-based cluster solution such as Kubernetes. This means scaling performance and capacity both up and down on demand.
- CAS solutions can be application and container-aware, using metadata (such as tags) to assist in data placement and protection.
- CAS solutions can easily be deployed in the public cloud, to offer abstraction from cloud-based storage services.

Performance testing storage

I/O performance is a key feature of any storage platform or medium. Historically, storage has been a bottleneck in computing systems because the relative performance of external peripherals is much slower than that of the processor or system memory.

Significant improvements in I/O performance have been made in recent years through the adoption of new protocols like NVMe and solid-state media, specifically NAND flash. Shared storage systems need to offer a high degree of efficiency to ensure flash and other media are used most effectively.

Metrics

Three metrics are typically used as a benchmark for storage performance.

- Latency – a measure of the time taken to complete a single I/O operation, measured from the perspective of the application. Latency is usually quoted in milliseconds or microseconds. Lower figures are better.
- Throughput – the capability of a storage system to process I/O requests, typically quoted in IOPS or I/Os per second. Higher IOPS values are better.
- Bandwidth – the amount of data a device or system can transfer over any given period of time, typically measured in MB/s (megabytes per second) or GB/s (gigabytes per second). Larger values are better.

Storage media vendors typically measure latency and IOPS using small block sizes (4KB) with a low queue depth, whereas bandwidth is usually measured with large block sizes and large queue depths. Both of these configurations maximise the capability of a storage device, offering a "best case" set of results rather than real-world metrics. Storage performance testing generally uses a mix of workload types, simulating typical I/O profiles.

Test methodology

In preparation of this report, four of the most popular CAS solutions were tested on a common set of high performance hardware. The platforms tested were:

ondat	**Ondat** commercial software from Ondat.
OpenEBS	**OpenEBS** an open-source initiative initially developed and supported by MayaData.
ROOK ceph	**Rook/Ceph** an open-source solution that uses Ceph as the storage platform and Rook as a management and presentation interface.
	Longhorn an open-source storage solution developed by Rancher Inc, now part of SUSE.

Each storage solution uses the Container Storage Interface (CSI) to provide normalised connectivity between applications and storage. CSI ensures that requests for storage volumes are implemented independently of the underlying process used by the storage platform.

Test Bed Configuration

Architecting IT performed a series of benchmark tests using internal lab equipment. The test bed consists of the following 4-server configuration:

- Dell R640 Server with 64GB DDR4-2400 DRAM
- Dual Intel Xeon Silver 4108 CPUs at @1.80Ghz
- One WD Gold NVMe SSD (960GB capacity) data disk
- One 136GB RAID-1 HDD pair (O/S drive, hardware RAID)
- 10GbE Networking
- Ubuntu Server 20.04

Testing was performed on a Kubernetes cluster with the vendor software installed. The configuration of each test is based on one Kubernetes master server and three Kubernetes data servers. The master server does not run any data services.

On each test scenario, the entire NVMe SSD was presented to the storage software and configured for use. This process followed the vendor's recommended practice, either pre-mounting an SSD as a file system or adding the SSD in the GUI after software installation.

Test Software

The test software uses open source fio to perform load testing on a single logical volume presented to the cluster. The range of tests is similar to those in the dbench suite but with some modifications.

- Ramp-up time is increased to 10 seconds, to counter any lazy write process when writing to a newly created volume.
- Run time is extended to 30 seconds to counter caching in the NVMe SSD.
- Each test allocates a 300GB volume and runs with a 250GB random dataset. The volume size chosen is deliberately larger than RAM available on each worker server.

The test script is available online here. Note that the fio script runs the following tests as independently executed commands, so it's not directly possible, for example, to compare latency figures against throughput or IOPS as they are from two separate test runs.

1. **Read IOPS –** block size of 4KB and queue depth of 16
2. **Write IOPS –** block size of 4KB and queue depth of 16
3. **Read bandwidth–** block size of 128KB and queue depth of 16
4. **Write bandwidth–** block size of 128KB and queue depth of 16
5. **Read latency –** block size of 4KB and queue depth of 4
6. **Write latency –** block size of 4KB and queue depth of 4
7. **Sequential Read –** block size of 1MB and queue depth of 16
8. **Sequential Write –** block size of 1MB and queue depth of 16
9. **Mixed Read/Write –** block size of 4KB and queue depth of 16

Prechecks

All four servers involved in the testing are configured with identical software releases and latest Ubuntu patches. At installation time, each server was validated with pre-installation checks and scripts provided by each vendor.

- **Ondat** – pre-requisites - https://docs.ondat.io/docs/prerequisites/
- **OpenEBS** – pre-requisites - https://docs.openebs.io/docs/next/prerequisites.html
- **Rook/Ceph** – pre-requisites - https://rook.io/docs/rook/v1.5/k8s-pre-reqs.html
- **Longhorn** – pre-requisites - https://longhorn.io/docs/1.1.0/deploy/install/

Platforms that require a key/value store such as etcd to maintain state will deploy that cluster software within the same Kubernetes cluster running the tests. This is not a recommended production configuration but done in this instance for simplicity.

The following release/versions were tested in this report:

- **Ondat – version 2.2**
- **OpenEBS – version 2.4**
- **Rook/Ceph – version 1.4**
- **Longhorn – version 1.0.2**

Version releases change frequently, and this report reflects the latest GA software release available at the time the testing was performed (late 2020).

Ondat

Ondat is a commercial software solution from Ondat. The software can be installed manually or through a scripted process as documented in the self-evaluation guide. Once installed, Ondat requires a valid licence to be assigned within 24 hours. A freemium developer licence is available for capacities of up to 5TB. Ondat offers GUI and CLI management.

OpenEBS

OpenEBS is an open-source solution that was originally developed by MayaData. The software is more complex to install than other offerings in that the user has to choose from multiple storage engines that offer varying levels of replication, performance and resiliency to an application. In this test scenario, we used the cStor engine as that provides replication across nodes in a similar configuration to the other products in this evaluation.

Rook/Ceph

Rook is an open-source gateway solution that maps software-defined storage into volumes that can be consumed by Kubernetes. Rook supports Ceph, EdgeFS, Cassandra, CockroachDB, NFS and YugabyteDB storage sources. In this test we used a Ceph configuration based on a standard testing script that uses three nodes to distribute the Ceph OSDs (object storage daemons).

Longhorn

Longhorn has been developed by the team that built Rancher, an open-source Kubernetes cluster management tool. Longhorn is an open-source distributed block-storage solution and CNCF Sandbox project. The software provides a GUI management interface.

Test Specifics

The aim of the testing process is to compare each of the four storage solutions and evaluate performance characteristics. The testing uses four configuration types:

1. **Local Volume (no Replica)** – the most basic of configurations with a single replica running on the same Kubernetes host as the fio script. This test demonstrates local device performance and is independent of any networking effects.
2. **Local Volume (with Replica)** – A local volume with replica on another host. This test introduces the effects of networking. We should expect to see increased latencies and lower throughput in this test, compared to the Local Volume test.
3. **Remote Volume (no Replica)** – in this test, the fio script runs on a different host to the primary host used to create the storage volume. As a result, we expect to see network effects similar to test 2, as I/O has to traverse the network and back.
4. **Remote Volume (with Replica)** – this test repeats test 3 but adds a replica volume on a third host that is neither the node running the fio test nor the node hosting the primary volume.

In each of the test scenarios, a Kubernetes StorageClass was created that provided the local/remote and replica configurations. The exception to this is with the Rook/Ceph test, where Ceph by default disperses data across all three nodes. In this instance, the local/remote tests aren't valid and only include the two replica/no replica test scenarios. Each testing scenario was run multiple times to check consistency, although the data from the first run is the only set of data documented in the results.

In order to validate the testing process, the creation of a container, volume and execution of the script were not automated. The script was executed manually for the following reasons:

- To ensure that any lazy write process on file system creation didn't affect the performance figures.
- To permit the validation of the container<->volume mapping and check the volume was the requisite size.
- To run iostat, top and iftop on each worker node. The output from these commands is not included but used to validate that there are no bottlenecks in CPU and network utilisation that could affect performance during test runs.

Once the test runs completed, the data from the fio output and the summary was collected and entered into a spreadsheet used to tabulate the results.

Test Results & Conclusions

The following graphs and annotations show the results of the testing and interpret the results in context of the tests being performed.

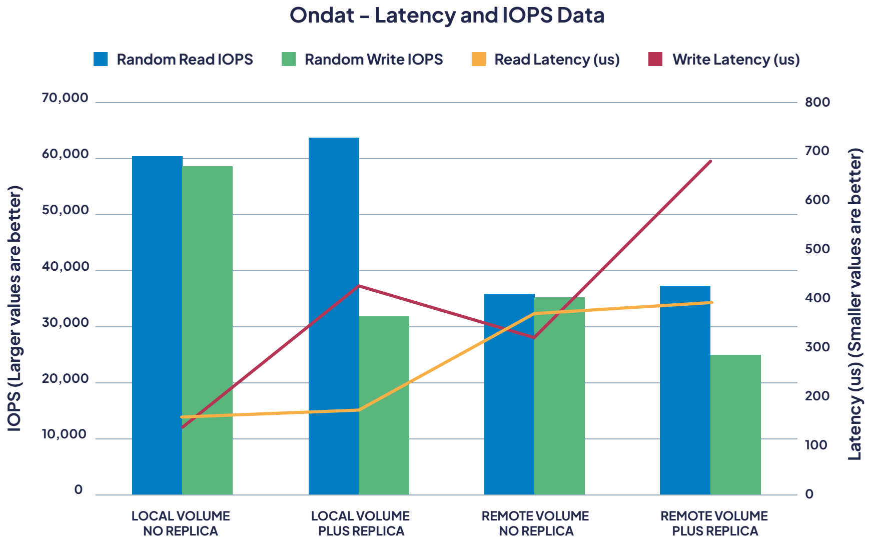
Solution-Specific Results

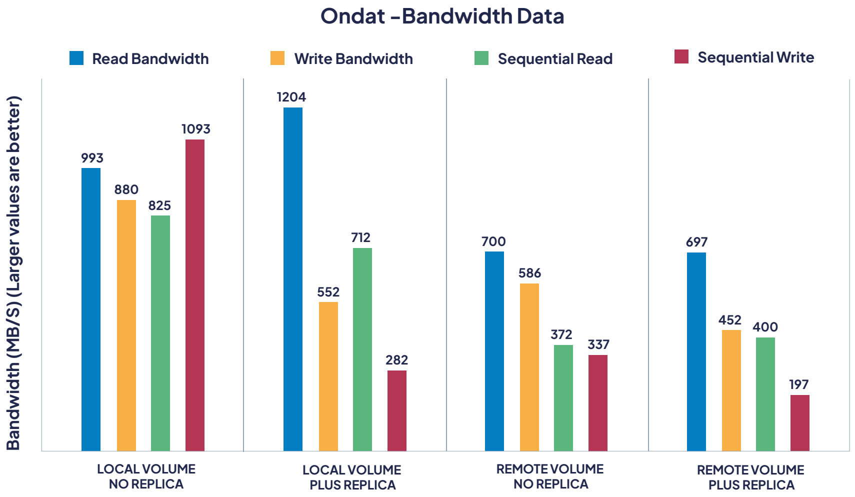
The results of individual testing with each storage solution are shown here first.

Ondat

The following graphs show data and analysis for the Ondat tests. The first graph shows latency (line graph with the scale on the right) and IOPS (bars with the scale on the left). Blue shades represent read I/O and green represent write I/O). Each pair of bars represents one of the four tests. For IOPS data, larger values are better, whereas for latency, smaller values are better.

The second graph (right) shows bandwidth data for each of the four tests. The tests pair read & write bandwidth in blue shades, followed by sequential bandwidth in green shades. In each case, larger values are better.





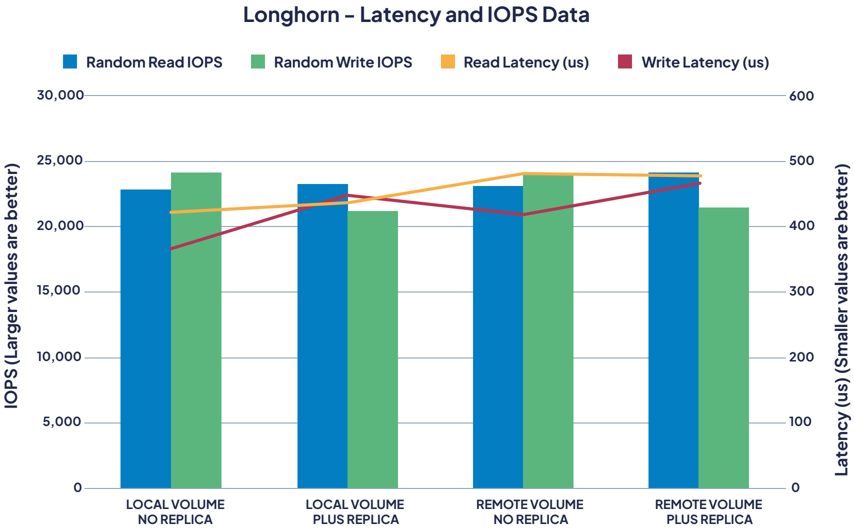
Observations:

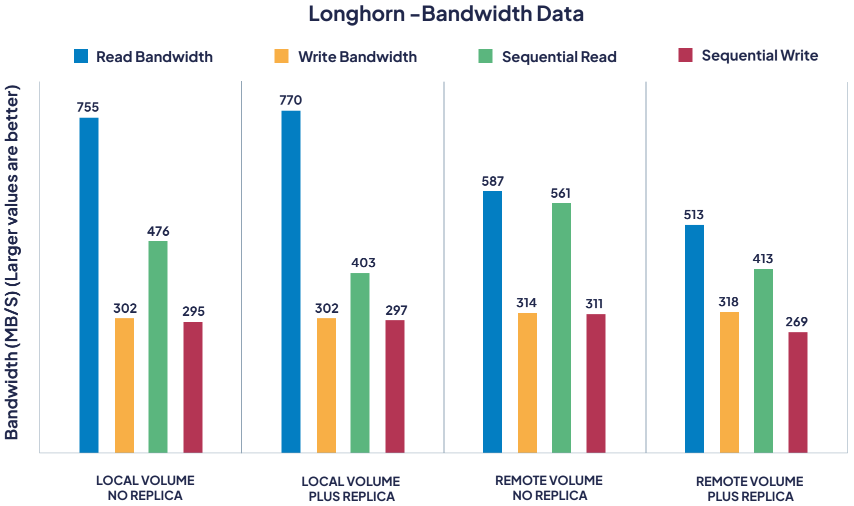
- The data shows consistent performance on read/write for a local volume.
- Local Volume plus Replica reduces the write I/O throughput, as two writes are made for every fio I/O.
- Remote Volume (no Replica) shows reduced throughput due to the impact of writing across the network.
- Remote Volume (plus Replica) shows further reduced write I/O as the test writes to two remote volumes on the network.
- Both read and write latency rise predictably as testing introduces networking effects.
- Ondat performs consistently well for standard and large-block I/O with local volume and no replica.
- Bandwidth increases with a replica volume for reads, presumably due to the ability to read from either mirror.
- Performance is consistent with remote volumes, with or without replicas.

Longhorn

The following graphs show data and analysis for the Longhorn tests. The first graph shows latency (line graph with the scale on the right) and IOPS (bars with the scale on the left). Blue shades represent read I/O and green represent write I/O). Each pair of bars represents one of the four tests. For IOPS data, larger values are better, whereas for latency, smaller values are better.

The second graph (right) shows bandwidth data for each of the two tests. The tests pair read & write bandwidth in blue shades, followed by sequential bandwidth in green shades. In each case, larger values are better.





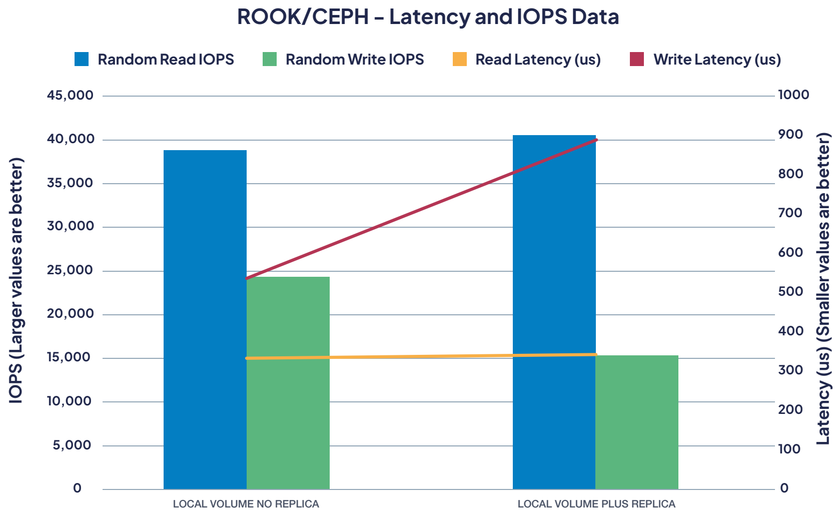
Observations:

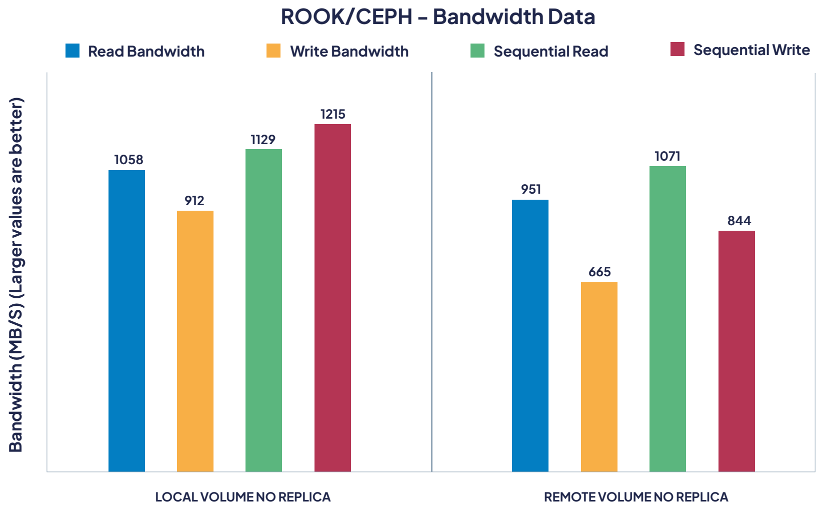
- The data for Longhorn testing shows marginally better write than read performance when writing locally.
- Remote performance is similar to the performance achieved with a local volume.
- Performance with a replica is similar to that without a replica.
- Write performance is consistent, irrespective of the block size and volume locality.
- Read performance is not improved with a replica (and diminishes with a replica).

Rook/Ceph

The following graphs show data and analysis for the Rook/Ceph tests. The first graph shows latency (line graph with the scale on the right) and IOPS (bars with the scale on the left). Blue shades represent read I/O and green represent write I/O). Each pair of bars represents one of two tests (as explained earlier, Ceph distributes data across all nodes, so aren't directly comparable with the other solutions). For IOPS data, larger values are better, whereas for latency, smaller values are better.

The second graph (right) shows bandwidth data for each of the two tests. The tests pair read & write bandwidth in blue shades, followed by sequential bandwidth in green shades. In each case, larger values are better.





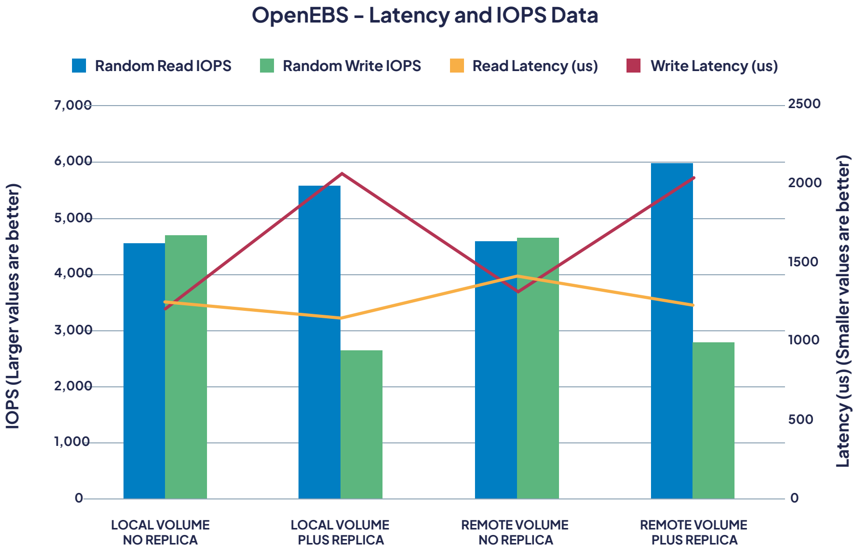
Observations:

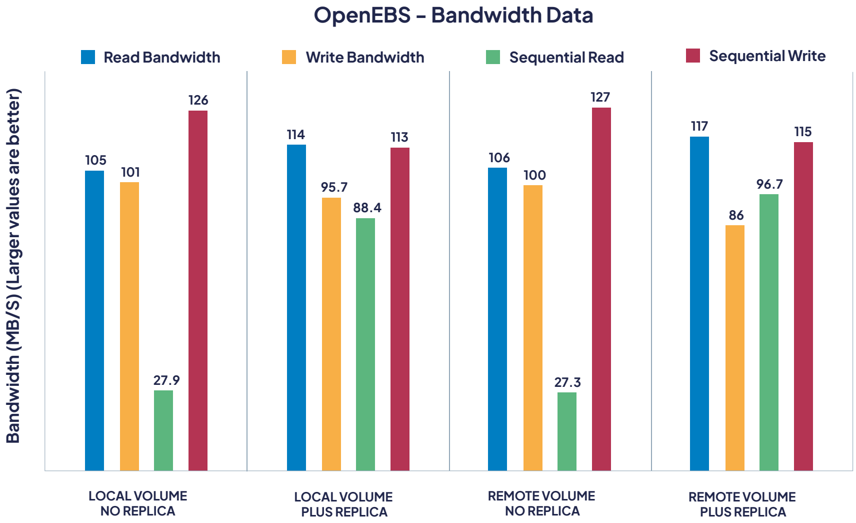
- Rook/Ceph testing shows only two data points, Local Volume (No Replica) and Local Volume (Plus Replica) as Ceph distributes data across all three storage nodes by default.
- Read latency is consistent in both tests, whereas write latency increases due to the additional networking effects.
- Read performance is significantly better than write performance.
- Write performance is heavily impacted by introduction of networking effects with a replica, despite the availability of three nodes and SSDs during the test.
- Data shows consistent I/O with for both I/O block sizes and for read and write activity. This is expected as data is spread across three nodes.
- The Local Volume plus replica test shows lower figures than just a local volume, as more I/O is now spread across the same infrastructure.

OpenEBS

The following graphs show data and analysis for the OpenEBS tests. The first graph shows latency (line graph with the scale on the right) and IOPS (bars with the scale on the left). Blue shades represent read I/O and green represent write I/O). Each pair of bars represents one of the four tests. For IOPS data, larger values are better, whereas for latency, smaller values are better.

The second graph (below) shows bandwidth data for each of the four tests. The tests pair read & write bandwidth in blue shades, followed by sequential bandwidth in green shades. In each case, larger values are better.





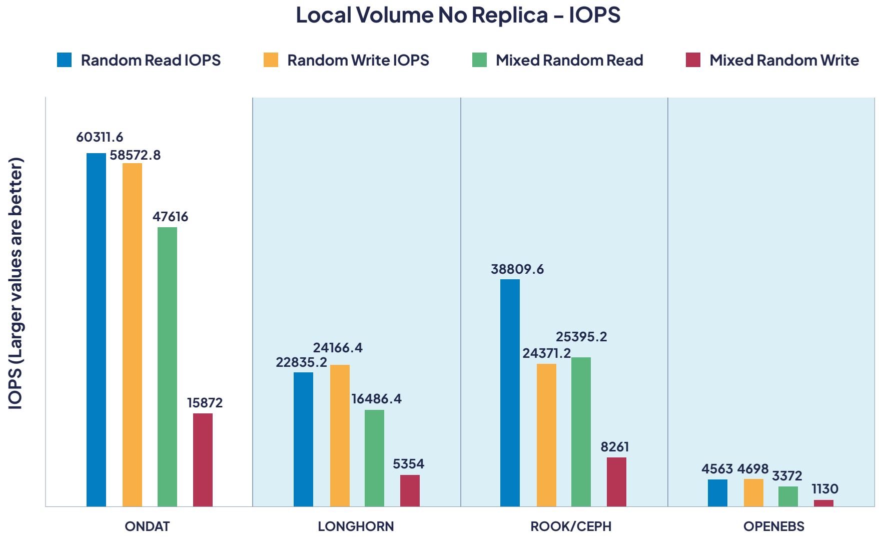
Observations:

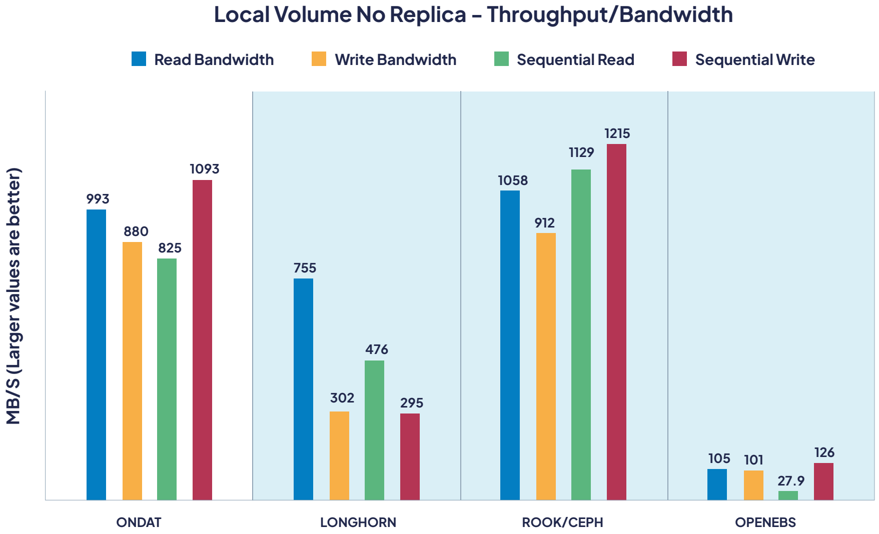
- OpenEBS testing using the cStor storage engine show similar read/write performance with a single local volume and no replica.
- Write performance is reduced with a remote replica, due to the impacts of networking. This is reflected in increased write latency.
- A remote volume performs about as well as a local volume.
- A remote volume with replica performs about as well as a local volume with replica.
- OpenEBS performance figures are much lower than the other solutions in this test, while latency is much higher. The result is that performance looks similar for local and remote volumes because the software latency is so high.
- OpenEBS performed poorly compared to other solutions in this test.
- OpenEBS performance on non-replica sequential reads was particularly poor.

Local Volume No Replica

Comparative Results

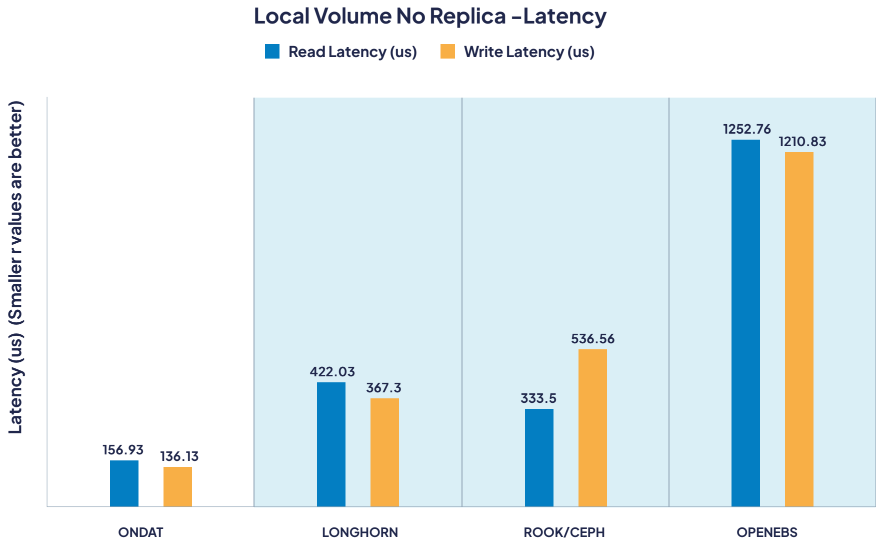
The results from each test are shown here in comparative form. Each graph lists data for the four platforms, in four sections (local/remote volumes with/without replica) and with three comparison points each (IOPS, Bandwidth and Latency). The data for Rook/Ceph is only partially shown in the Local Volume graphs as data is spread across all nodes.



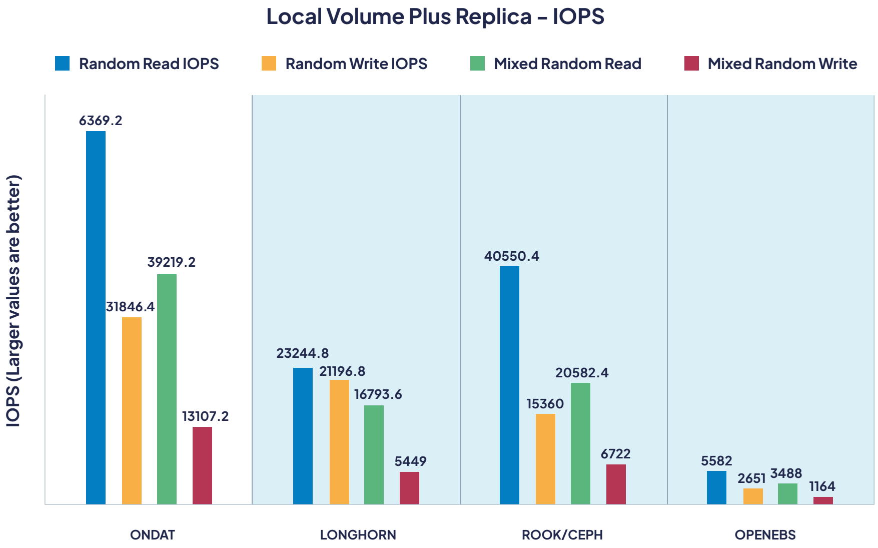


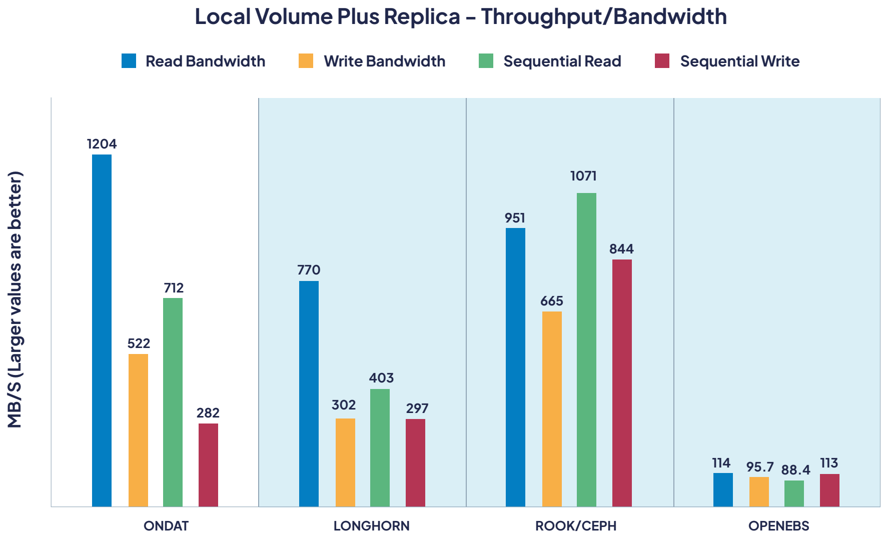
Observations:

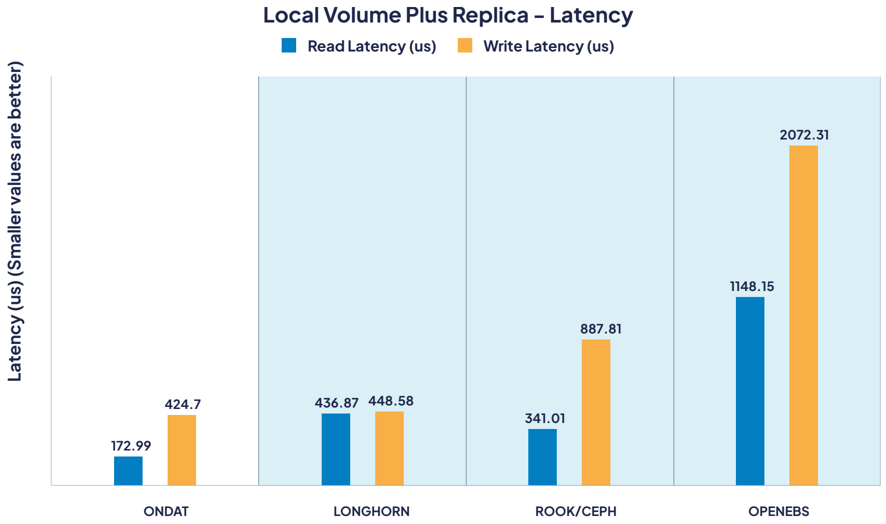
- Ondat performs strongly in these test results in comparison to Longhorn, Rook/Ceph and OpenEBS.
- Rook/Ceph performs well on the throughput test but has access to 3x the storage capacity and media bandwidth of the other solutions.
- OpenEBS performed poorly all of the test results, which seems to be directly a result of much higher I/O latency than the other solutions.



Local Volume Plus Replica





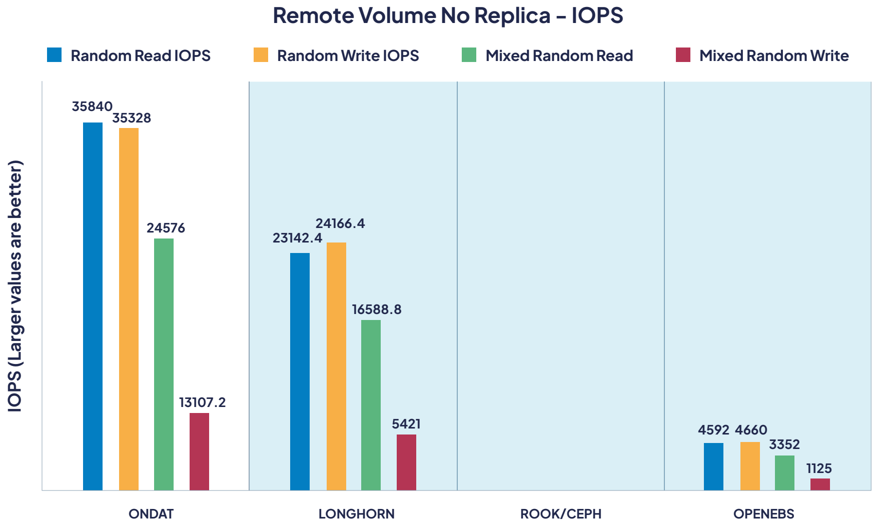


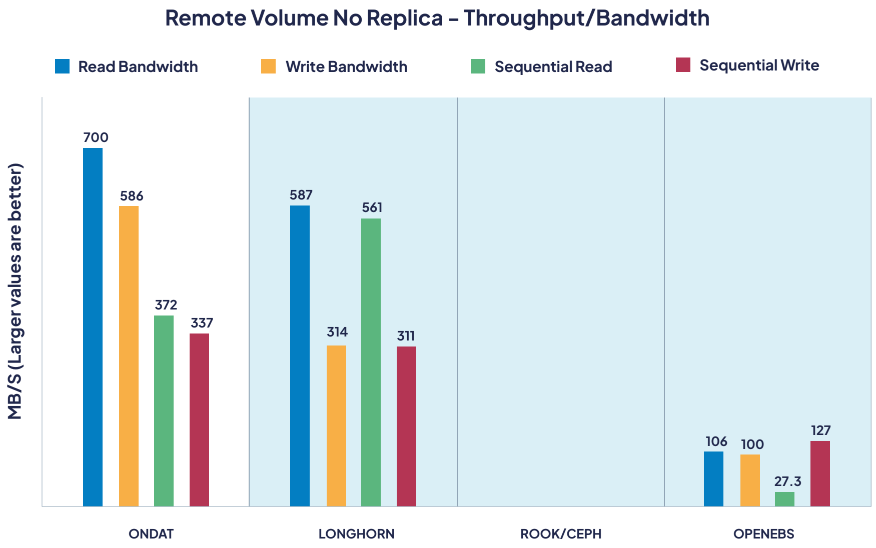
Observations:

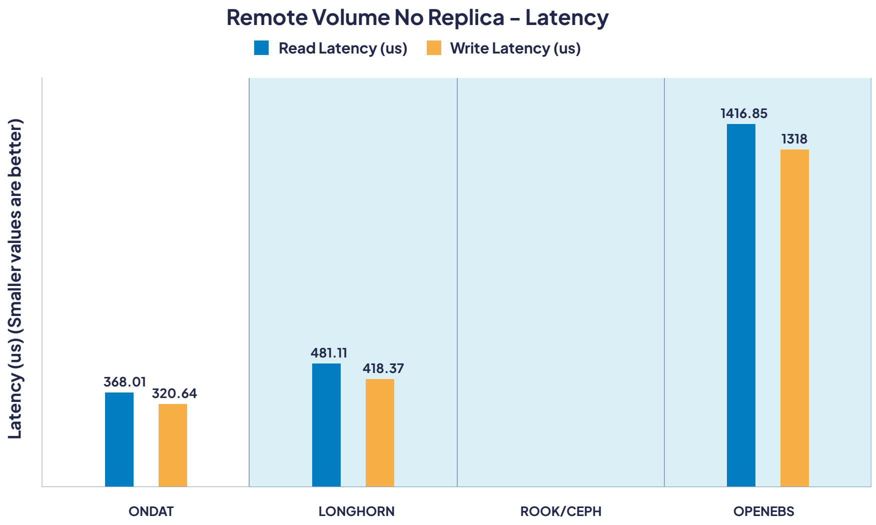
- Ondat performed best in these tests with the exception of the sequential I/O figures.
- Rook/Ceph performed well but has access to 3x the IOPS and the bandwidth across three nodes.
- OpenEBS was again the worst performer.
- Longhorn latency figures show no difference in read/write I/O, which indicates a lack of read caching.

Remote Volume No Replica

In these tests, no data is presented for Rook/Ceph as previously discussed.





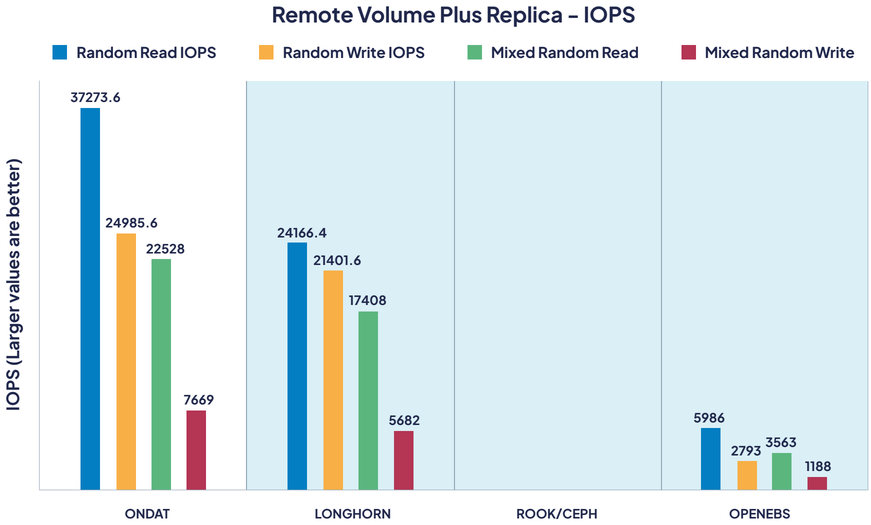


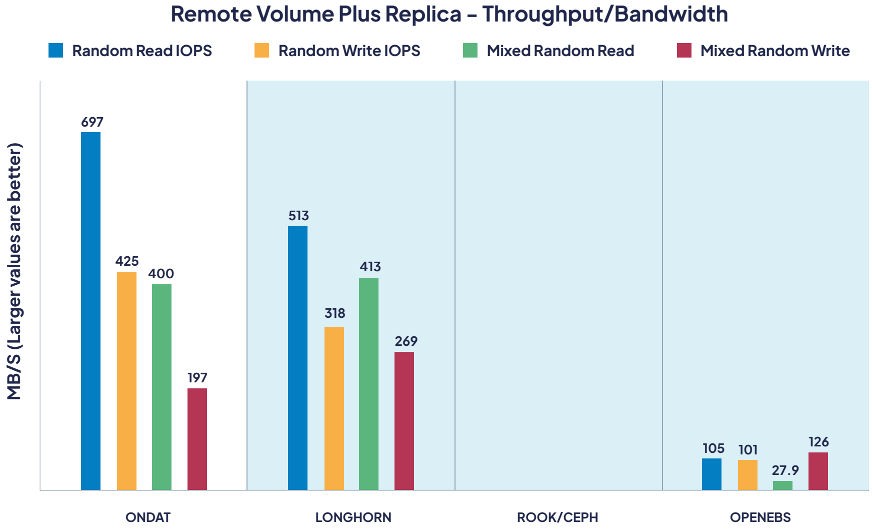
Observations:

- Ondat is the strongest performer in the tests.
- Longhorn performs well, clearly taking advantage of remote read to improve performance.
- Again, OpenEBS performed poorly, specifically on remote sequential read I/O.

Remote Volume Plus Replica

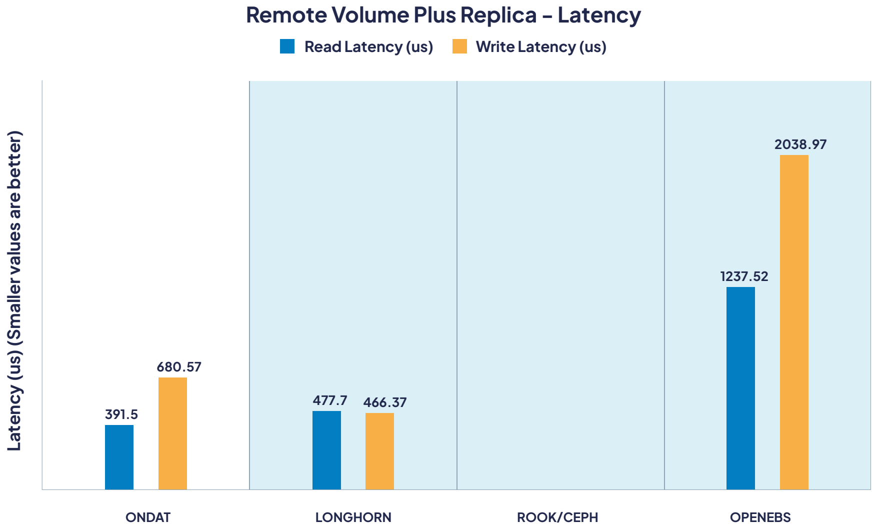
In these tests, no data is presented for Rook/Ceph as previously discussed.





Observations:

- In this test, Ondat performs better than the other solutions, except on remote write bandwidth. This exception is also demonstrated in the higher latency figures for Ondat compared to Longhorn.



Conclusions

Taking all of the test results into consideration, Ondat was the best overall performer in all categories. Ceph performed well in some tests, but in this configuration had the benefit of aggregating bandwidth and throughput across three times the number of drives and storage nodes during the entire testing.

Longhorn offered an acceptable level of performance, but clearly doesn't make use of caching to improve read I/O. This is a missed opportunity, especially in containerised environments where the ability to deliver I/O from RAM will significantly improve application performance.

OpenEBS turned out to be the worst performer. The configuration tested uses cStor to provide a fair comparison in terms of replication at the storage layer. The current implementation of cStor could be the Achilles' heel of the OpenEBS implementation, as separate testing with alternative storage engines produced much better results. The alternative tests are not documented here but were performed to ensure the OpenEBS configuration had been implemented correctly.

Choosing the right storage platform for applications is a combination of usability, performance and reliability. In this test series, it is clear that Ondat is the best choice of product where performance is a key requirement. Performance across all metrics exceeded that of other vendor solutions.

Further Testing

Performance testing could be described as an art rather than a science. While benchmarks can give a good indication of performance capabilities between solutions, further testing could dive deeper into application specific performance. This could look at transactional, rather than I/O-based performance numbers and is likely to be more important as we see further containerisation of transactional databases and other applications.

architectingit

More Information

Architecting IT is a brand name of Brookend Ltd, an independent consultancy, working to explain technology and business value to the end customer. This report was commissioned by Ondat, however editorial control of the final published document remains with Brookend Ltd.

Email: architectingit@brookend.com

Twitter: @architectingit

The Author

Chris M Evans has worked in the technology industry since 1987, starting as a systems programmer on the IBM mainframe platform. After working abroad, he co-founded an Internet-based music distribution company during the .com era, returning to consultancy in the new millennium. Chris writes a popular blog at **https://www.architecting.it/blog**, co-hosts the Storage Unpacked podcast, attends many conferences and invitation-only events and can be found providing regular industry contributions through Twitter **(@chrismevans)** and other social media outlets.

Get started today with Ondat

Visit us at www.ondat.io or email us at info@ondat.io