

BENCHMARKING POSTGRES SQL STORAGE PERFORMANCE ON KUBERNETES

A Performance Analysis Using Percona Operator
for PostgreSQL



Executive Summary

We benchmarked the performance of Percona Operator for PostgreSQL on Kubernetes, testing various storage options including [OpenEBS](#), [Rook](#), [Lightbits](#), [Portworx](#), and [AWS EBS volumes](#). The goal is to compare performance across different storage solutions, focusing primarily on I/O operations.

Two sets of tests were conducted: one with a less-tuned database configuration to force more physical disk I/O and another with a well-tuned database relying more on shared buffers. Fio (Flexible I/O Tester) based kbench and pgbench tools were used to gather performance metrics such as IOPS, bandwidth, latency, and transactions per second (TPS).

The results highlight that OpenEBS using built-in local storage and Portworx generally performed better than other options in terms of I/O-intensive workloads, particularly with random reads and writes. Lightbits showed improved performance at higher scale factors. When the database was well-tuned, differences in storage performance narrowed, as the shared buffers reduced direct disk I/O.

Environment

Kubernetes: AWS EKS version 1.29.4

Instances: **i4i instances** to use local storage

Network: all nodes in the same subnet

Database

We are running Percona Operator for PostgreSQL version 2.3.0 that deploys PostgreSQL version 16.3. Even though Operator deploys pgBouncer for connection pooling, we are not using it in our tests to show the client/thread values and load.

For all tests we deploy 1 primary and 2 replicas running on dedicated Kubernetes nodes.

Tooling

We used **kbench** (fiio based) and **pgbench** to perform the tests. We used both to ensure we evaluate the storage from both a raw performance perspective and how it functions within the context of a database system.

Storage

We benchmarked the following storage options:

- AWS EBS: Cloud-based block storage service for EC2 instances, offering persistent storage volumes.
- OpenEBS: Open-source container-attached storage solution, providing persistent storage for stateful applications on Kubernetes.
- Rook: Open-source cloud-native storage orchestrator, facilitating deployment and management of distributed storage systems on Kubernetes.
- Portworx: Enterprise Kubernetes storage platform, offering persistent storage, data management, and disaster recovery for containerized applications.
- Lightbits: Software-defined storage solution, providing high-performance block storage over standard Ethernet networks using NVMe/TCP.

Key points about the storage configurations:

- All storage options, except AWS EBS, are configured to use NVMe flash drives exclusively.
- AWS EBS is configured with io2 storage and provisioned IOPS of 64,000 (although this limit wasn't reached during testing).
- The Lightbits cluster consists of three i4i.8xlarge instances.
- To minimize latency, storage replication was avoided whenever possible. While this isn't ideal for production, our goal was to measure maximum performance.

Tests

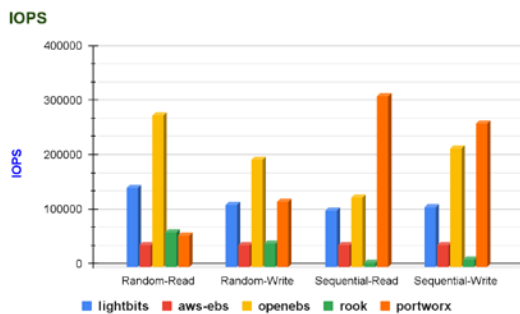
We ran 3 different tests:

Test name	Pure storage I/O	Force disk IO for PostgreSQL	Shared buffers
Description	No PostgreSQL, just pure storage test with kbench	Use less RAM and shared buffer to force more I/O with the storage	Increase the use of shared buffers to show how it affects the storage I/O
PostgreSQL compute	N/A	12 vCPU 10 GB RAM	12 vCPU 100 GB RAM
Storage	1024 GB PVCs	1024 GB PVCs	1024 GB PVCs
PostgreSQL configuration	N/A	PostgreSQL configuration max_wal_size: 1GB shared_buffers: 1GB effective_io_concurrency: 100 max_connections: 500	PostgreSQL configuration max_wal_size: 25GB shared_buffers: 25GB effective_io_concurrency: 100 max_connections: 500

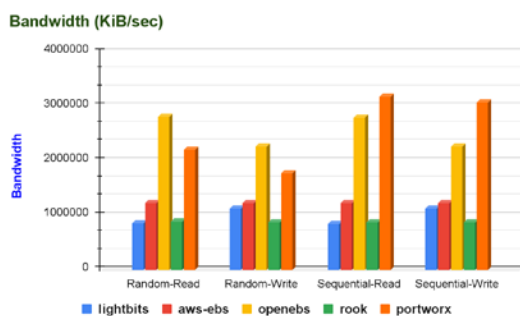
Test results

Test 1 - Pure storage I/O

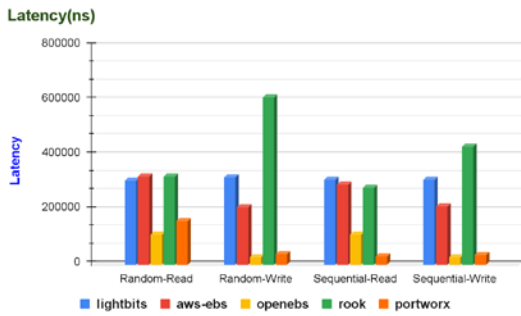
For this test we used only kbench. We were looking at three parameters: IOPS, Bandwidth and Latency.



- Higher IOPS indicates better performance.
- OpenEBS provided the best IOPS in random read and write.
- Portworx provided the best IOPS in Sequential read and write.



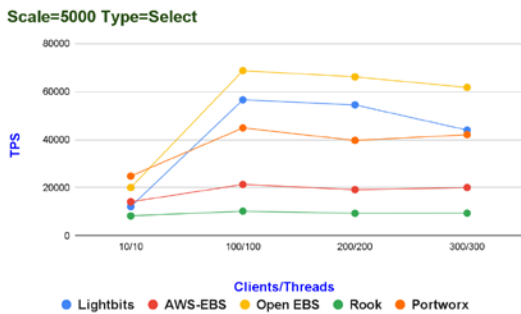
- Higher bandwidth indicates better performance.
- OpenEBS provided the best Bandwidth in random read and write.
- Portworx provided the best Bandwidth in Sequential read and write.



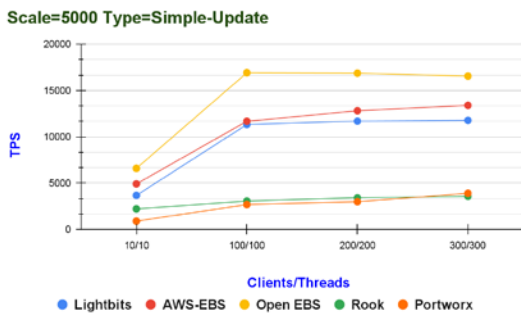
- Lower Latency indicates better performance.
- OpenEBS provided the least latency in Random reads, Random writes and Sequential writes. Only Portworx had lesser latency than OpenEBS in Sequential reads.
- Rook had the highest latency in writes and also had relatively high latency in reads.

Test 2 – Force disk IO for PostgreSQL

This is where we add PostgreSQL into the tests and force disk IO by deliberately tuning PostgreSQL to reduce shared buffers usage. We ran pgbench with two built-in tests: select and simple-update. We also changed scale factor (indicates the number of rows created for the pgbench tests) and number of **clients and threads** (to add concurrency). We noticed that scale factor does not significantly impact the results. We measured performance through Transactions Per Second (TPS).



- OpenEBS had the best performance once the number of clients and threads crossed 100. Rook had lesser performance compared to other storages.



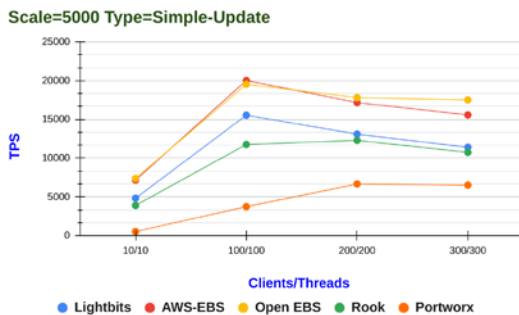
- OpenEBS had the best performance. Rook had lesser performance compared to other storages.
- Portworx had lesser performance than other storages until the number of clients reached 200, after which the TPS dropped for rook compared to Portworx.

Test 3 – Shared buffers

We increase the resources allocated to PostgreSQL containers (from 10GB to 100GB memory) and tune shared buffers. As expected, the storage option does not drastically impact the results.



- AWS EBS, OpenEBS and Lightbits showed similarly high performance.
- Portworx on the other hand was at the bottom this time.



- AWS EBS and OpenEBS showed similarly high performance.
- Rook and Lightbits were performing quite the same.
- Portworx showed lesser performance of all.



Have questions?

Percona's database experts are ready to assist with your PostgreSQL deployment on Kubernetes, offering specialized expertise in managing databases within containerized environments. Whether you need guidance, implementation, or consultation, we provide tailored solutions for setting up, optimizing, and scaling your Kubernetes-based databases to meet your specific project requirements.



Scan or click

MySQL, PostgreSQL, MongoDB, Valkey, and Redis.
Operated with experts across on-prem, cloud,
Kubernetes, and hybrid environments. No lock-in.

Take the next step

Contact us today to discover how Percona can help you optimize your database environment while meeting your cost and performance objectives.