Flash Memory Summit

Turbo-charging the Cloud with Zoned Storage

Dennis Maisenbacher
Principal, Emerging System Architectures
Western Digital

Image source: https://cdn.shopify.com/s/files/1/0444/5369/1553/products/2_d44a81cc-91b1-4de9-9089-9ce84d665943_1024x1024@2x.jpg?v=1604501611

Overview

- Cloud-Native Storage Challenges & Zoned Storage
- Work on Enabling Zoned Storage as a 1st Class Cloud Citizen
  - Longhorn
  - OpenEBS - Mayastor
  - spdk-csi - CSAL
- Showcase

Cloud-Native Storage Challenges & Zoned Storage

Cloud-Native Storage at Scale

- Cloud Service Providers (**CSPs**) are constantly **challenged** with **large volumes of data** and increasing **customer demand** for **cost-effective storage and high performance**.
  - Measurements: IOPS/TB, GB/$, TB/Rack
- To solve these challenges, CSPs provide **software-defined storage solutions** by exposing the **data as a service** that implements **multiple storage tiers** to provide the appropriate benefits.

CSP's Challenges – Storage Hardware

Storage tiers have inherent **physical characteristics**, which, if to be used in the cloud, need to be **effectively managed**

Flash-based SSDs (**Performance Tier**):

- Media type (TLC/QLC), WAF, DRAM, OP → **Cost (GB/$)**
- Relatively expensive compared to HDDs. The characteristics of conventional SSDs are such that they exhibit:
  - **Write amplification** which impacts **lifetime, performance**, and **overall cost**.
  - Typical **overheads** include **2x cost** (CacheLib paper) and **3-4x reduced lifetime** (rw workload *), reduced lifetime (contradicts the requirement of deploying capacity storage (i.e., QLC) for 5-7 years)
  - Bandwidth **penalties** in single device - **multi tenant** setups.

→**ZNS SSDs** solve these issues while also maintaining **high performance**.

HDDs (**Capacity Tier**):

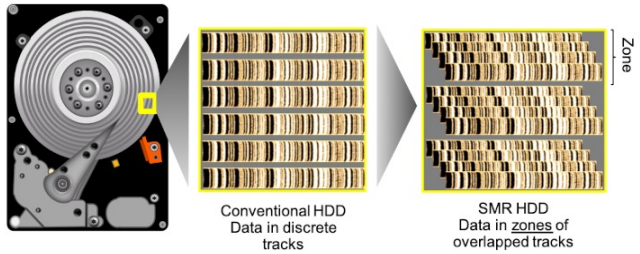
- Rotation speed, actuator and platter count →**Cost and Throughput per GB (IOPS/GB & GB/$)**
- **Large-scale HDD deployments** (power and space constrained) are **improved** using SMR technology (22TB vs 26TB)

*"Write amplification analysis in flash-based solid state drives" - Xiao-Yu Hu, Evangelos Eleftheriou, Robert Haas, Ilias Iliadis, and Roman Pletka

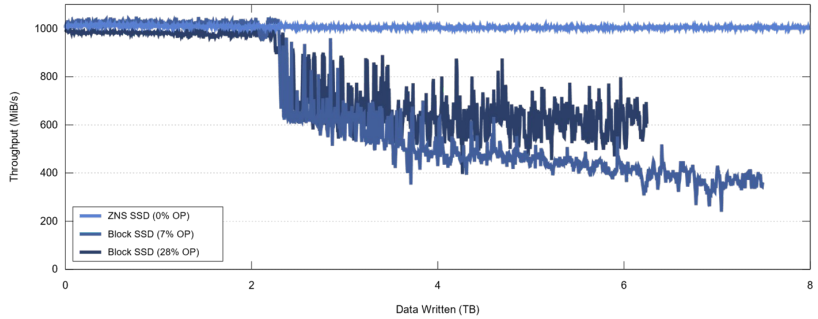
The Benefits of Zoned Storage

Flash Memory Summit

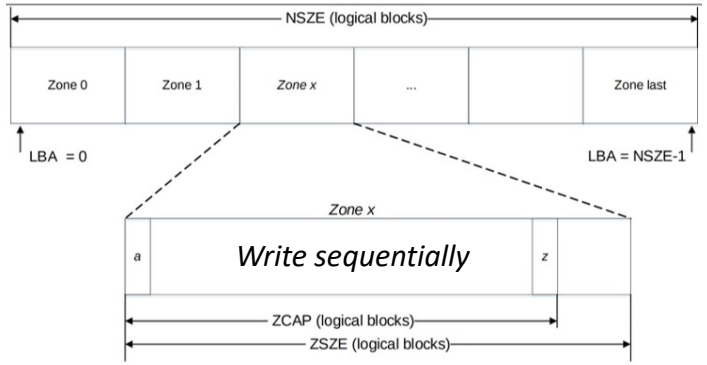
SMR HDDs offer 18% additional capacity with the newest 26TB SMR HDD over conventional CMR HDD.



SSDs with Zoned Namespace support offer 3-4x higher performance, 7-28% higher capacity, and usage of QLC in write-heavy workloads.



NVMe® Zoned Namespace



- Deployed heavily at hyperscalers and in large-scale storage clusters (SMR HDDs and ZNS SSDs)
- To take advantage of SSDs with **Zoned Namespace** support and **SMR** HDDs, the software requires support

Image sources: zonedstorage.io and the NVM Express Zoned Namespace Command Set 1.1 specification

Todays Linux® Eco-System

Official Linux® support since 2016

- Zoned API from kernel version 4.10 (Feb 2017)
- ZNS support added in kernel version 5.9 (Oct 2020)
- UFS support to be available when standardized (~2023)

5+ Linux® Distributions with Zoned Storage Support

- RHEL 9+, CentOS 7+, Fedora 33+, Debian 11+, and Ubuntu 21.04+

Two File-systems supports Zoned Storage

- f2fs (client - UFS) and btrfs (enterprise - ZNS/SMR)

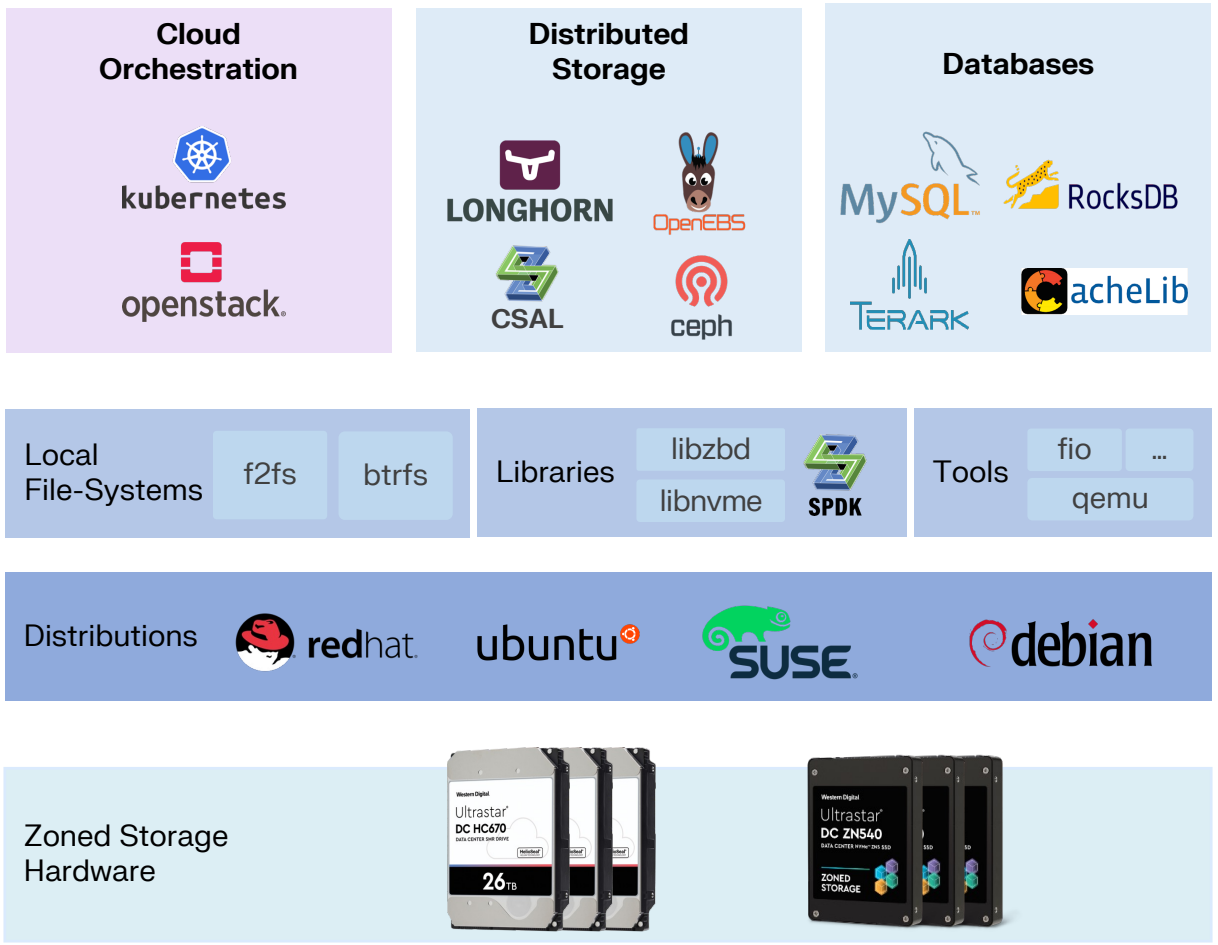
Library/Tools support

- libzbd, libnvme, SPDK, fio, qemu, blkzone, blktests, …

Broader Enablement

- Cloud Orchestration, Databases, Distributed Storage, Databases, Caching, Generic storage

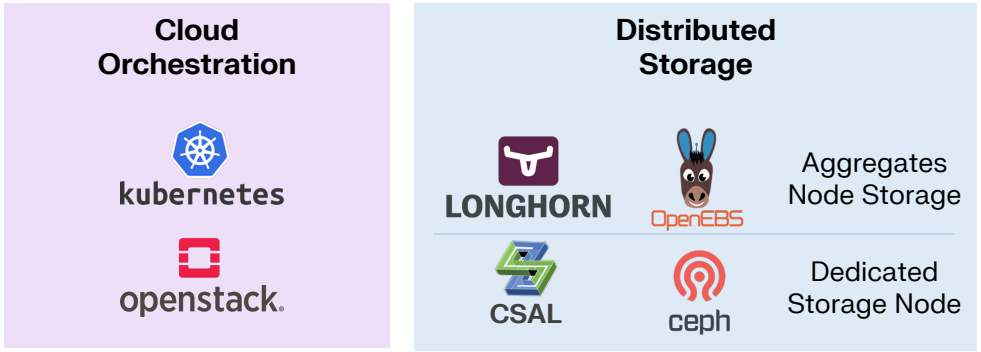
Mature, robust, and adopted by some of the biggest consumers of storage



Slide credits: Matias Bjørling

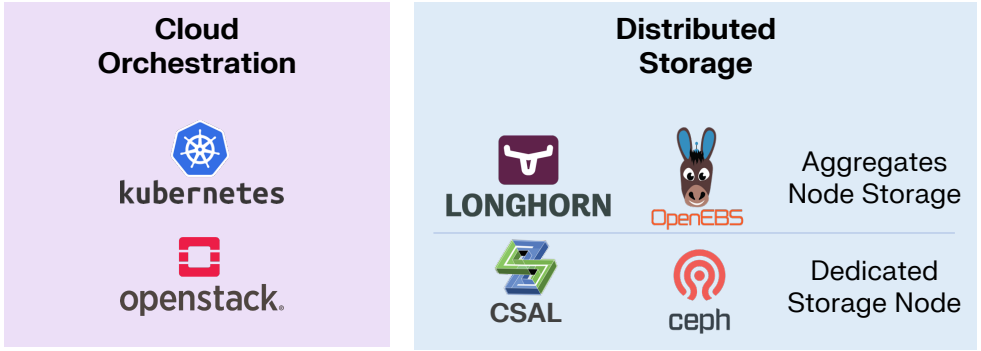
Previously Missing Link for Cloud Applications

- Tight integration with the Cloud Orchestration Platform required.
- The Storage as a Service is provided by the specific CSP (Azure, AWS, GCP) which implements internal solutions to gain the advantages.
- But how to get the benefits with on-premise clouds and/or hybrid clouds?



Previously Missing Link for Cloud Applications

- Tight integration with the Cloud Orchestration Platform required.
- The Storage as a Service is provided by the specific CSP (Azure, AWS, GCP) which implements internal solutions to gain the advantages.
- But how to get the benefits with on-premise clouds and/or hybrid clouds?



Enablements

- **Longhorn** and **CSAL** provide a **generic block device/file-system** abstraction for containers using zoned storage
- With **Mayastor**, we made it such that **zoned storage** devices can be **explicitly exposed** to a container!

→ Ceph - FMS Talk tomorrow by *Aravind Remesh* "**ZNS in cloud with Ceph Crimson-OSD**"

Zoned Cloud Native Storage

1) Longhorn – Zoned Cloud Native Storage

- Longhorn is a persistent distributed storage system with no single point of failure
  - A replica is accessing the SSD through the file system abstraction
  - Containers accesses a POSIX compatible filesystem as the k8s volume or optionally a conventional block device
- Changes necessary
  - Make use of the native support for zoned storage in BTRFS
    - Multi-year effort on enablement now makes it easy to integrate
    - Works, with minimal configuration, out of the box

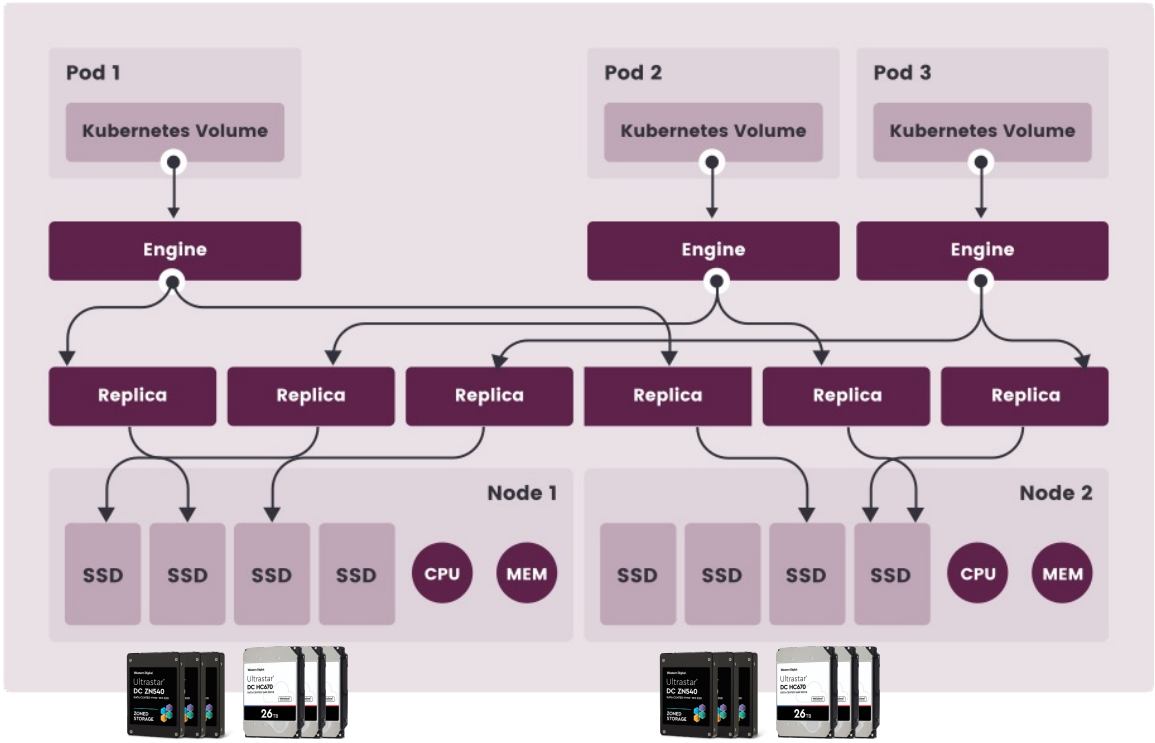


Image source: https://longhorn.io/img/diagrams/architecture/how-longhorn-works.svg

2) Mayastor – Zoned Cloud Native Storage

- OpenEBS - Mayastor (written in Rust 🦀) is much like Longhorn a persistent highly available Container Attached Storage (CAS) solution
  - Utilizes SPDK → Avoiding storage related kernel modules for node uptime
  - Ongoing plumping work in fixing up the Mayastor I/O path to natively handle ZNS NVMe block devices → First proposal with replica factor 1
  - Interesting challenges with zoned replication:
    - Keep zone write pointer across replicas in sync
    - What happens on I/O errors?
- CSI driver consumer will receive a zoned block device via NVMe-oF

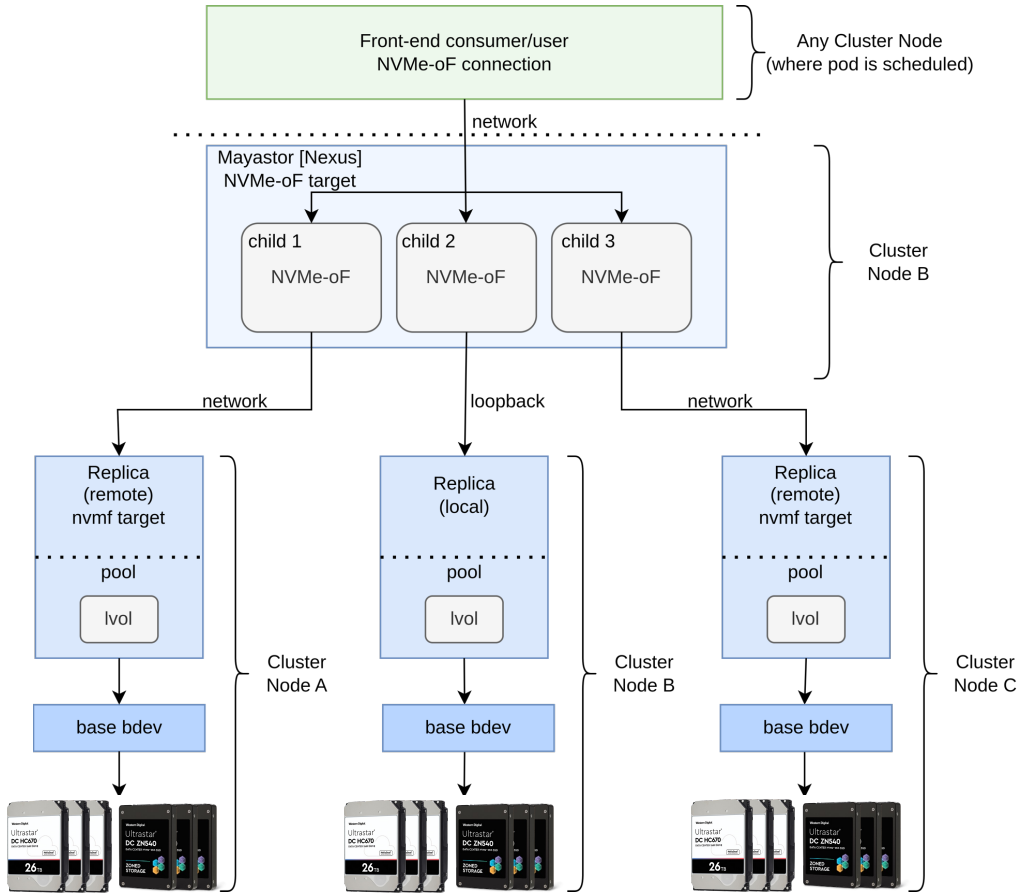
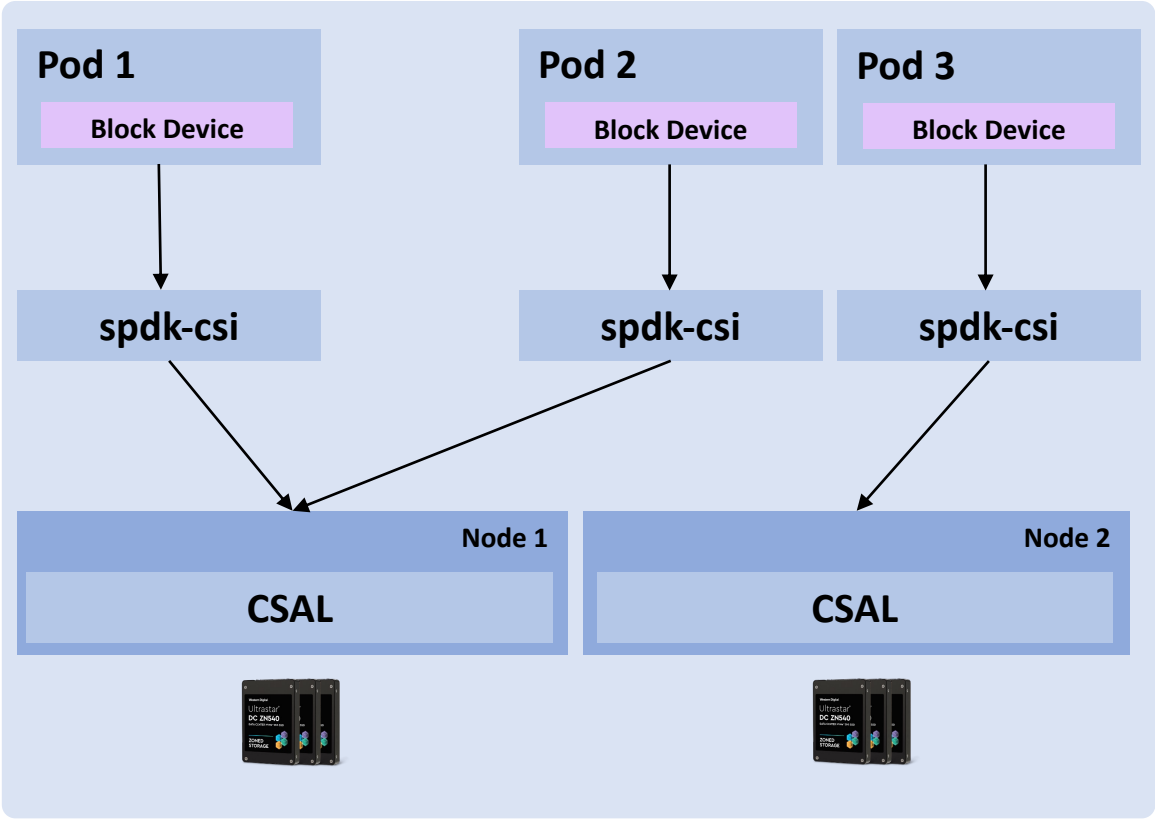


Figure inspiration: https://mayastor.gitbook.io/introduction/additional-information/i-o-path-description

3) SPDK CSAL – Zoned Cloud Native Storage

- The Cloud Storage Acceleration Layer (CSAL) which has WIP zoned storage support can be deployed as a CAS though the spdk-csi driver or Mayastor
  - Implements a caching and translation layer that transforms zoned storage to conventional storage
- CSAL uses a conventional (high-performance) block device for metadata and writes sequentially to the ZNS SSDs, thus hiding ZNS' sequential write constraint
- Exposed as a conventional block device over a NVMe-oF target

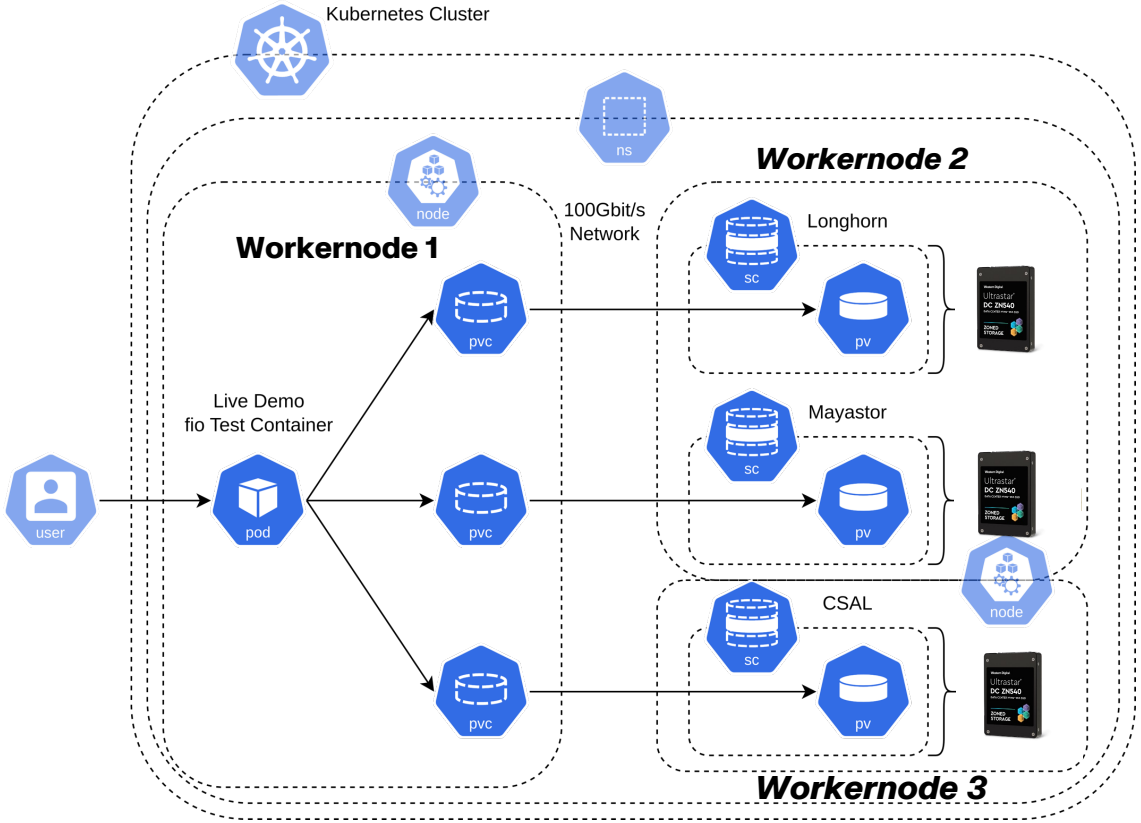


Showcase

Showcase Setup

Flash Memory Summit

- 4 Nodes (one control and three worker nodes)
  - Workernode 1
    - User workload - Container /w fio
  - Workernode 2
    - Longhorn - Exposes a generic file-system/block device backed by an Ultrastar® DC ZN540 SSD
    - Mayastor - Exposes an Ultrastar® DC ZN540 SSD natively to a container
  - Workernode 3
    - SPDK CSAL - Exposes a generic block device over NVMe-oF using an Ultrastar® DC ZN540 as its storage backend



Showcase Setup

Flash Memory Summit

```yaml
---
# FMS Demo Pod
apiVersion: v1
kind: Pod
metadata:
  name: fms-demo
spec:
  containers:
  - name: fms-fio
    image: dennis-k8s-masternode:32000/test/fms-fio-container:latest
    imagePullPolicy: Always
    command: [ "/bin/sleep" ]
    args: [ "365d" ]
    volumeDevices:
    - name: mayastor-zoned-bdev
      devicePath: /dev/nvme0n1
    volumeMounts:
    - mountPath: "/longhorn-vol"
      name: longhorn-vol
    - mountPath: "/spdkcsi-csal-vol"
      name: spdkcsi-csal-vol
  volumes:
  - name: longhorn-vol
    persistentVolumeClaim:
      claimName: longhorn-pvc
  - name: mayastor-zoned-bdev
    persistentVolumeClaim:
      claimName: mayastor-zoned-bdev-pvc
  - name: spdkcsi-csal-vol
    persistentVolumeClaim:
      claimName: spdkcsi-csal-pvc
```

Zoned Block Device

POSIX Filesystem on
Conv Block Device
Backed by Zoned Storage

```
ubuntu@dennis-k8s-masternode:~$ kubectl exec --stdin --tty fms-demo -- /bin/bash
root@fms-demo:/# lsblk
NAME    MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0     7:0    0  63.4M  1 loop
loop1     7:1    0 111.9M  1 loop
loop2     7:2    0  53.3M  1 loop
loop3     7:3    0 169.1M  1 loop
loop4     7:4    0   1.8T  0 loop
sda       8:0    0     8G  0 disk /longhorn-vol
vda     252:0    0   189G  0 disk
|-vda1  252:1    0 188.9G  0 part /etc/resolv.conf
|                                 /etc/hostname
|                                 /dev/termination-log
|                                 /etc/hosts
|                                 /spdkcsi-csal-vol
|-vda14 252:14   0     4M  0 part
`-vda15 252:15   0   106M  0 part
vdb     252:16   0     1M  0 disk
nvme0n1 259:1    0   1.8T  0 disk
nvme1n1 259:3    0    32G  0 disk
root@fms-demo:/# cat /sys/block/nvme0n1/queue/zoned
host-managed
root@fms-demo:/# fio fms-example.fio
longhorn-write: (g=0): rw=write, bs=(R) 64.0KiB-64.0KiB, (W) 64.0KiB-64.0KiB, (T) 64.0
...
spdkcsicsal-write: (g=0): rw=write, bs=(R) 64.0KiB-64.0KiB, (W) 64.0KiB-64.0KiB, (T) 6
...
mayastor-write: (g=0): rw=write, bs=(R) 64.0KiB-64.0KiB, (W) 64.0KiB-64.0KiB, (T) 64.0
...
longhorn-read: (g=1): rw=read, bs=(R) 64.0KiB-64.0KiB, (W) 64.0KiB-64.0KiB, (T) 64.0Ki
...
spdkcsicsal-read: (g=1): rw=read, bs=(R) 64.0KiB-64.0KiB, (W) 64.0KiB-64.0KiB, (T) 64.
...
mayastor-read: (g=1): rw=read, bs=(R) 64.0KiB-64.0KiB, (W) 64.0KiB-64.0KiB, (T) 64.0Ki
...
```

POSIX Filesystem on
Conv Block Device
Backed by Zoned Storage

Zoned Block Device

Example workload

```yaml
---
apiVersion: v1
kind: Pod
metadata:
  labels:
    app: percona-server
  name: percona-server
spec:
  initContainers:
  - name: zenfs-setup
    image: docker.io/library/percona:ps-8.0.33-25
    command: ['/bin/bash', '-c', 'rm -rf /tmp-zenfs/zenfs-aux-test && zenfs mkfs --zbd=nvme0n1 --aux_path=/tmp-zenfs/zenfs-aux-test --force']
    volumeMounts:
    - name: mysql-config-directory
      mountPath: /etc/my.cnf.d
    - name: tmp-directory
      mountPath: /tmp-zenfs
    volumeDevices:
    - name: zns-block-device
      devicePath: /dev/nvme0n1
  containers:
  - name: percona-server
    image: docker.io/library/percona:ps-8.0.33-25
    ports:
    - containerPort: 3306
      name: mysql
    volumeMounts:
    - name: mysql-config-directory
      mountPath: /etc/my.cnf.d
    - name: tmp-directory
      mountPath: /tmp-zenfs
    volumeDevices:
    - name: zns-block-device
      devicePath: /dev/nvme0n1
    env:
    - name: MYSQL_ROOT_PASSWORD
      value: "test"
    - name: INIT_ROCKSDB
      value: "yes"
  volumes:
  - name: mysql-config-directory
    hostPath:
      path: /home/debian/k8s-percona-mysql-config
      type: Directory
  - name: mysql-directory
    hostPath:
      path: /home/debian/k8s-percona-mysql
      type: Directory
  - name: tmp-directory
    hostPath:
      path: /home/debian/k8s-percona-tmp
      type: Directory
  - name: zns-block-device
    persistentVolumeClaim:
      claimName: block-device
```

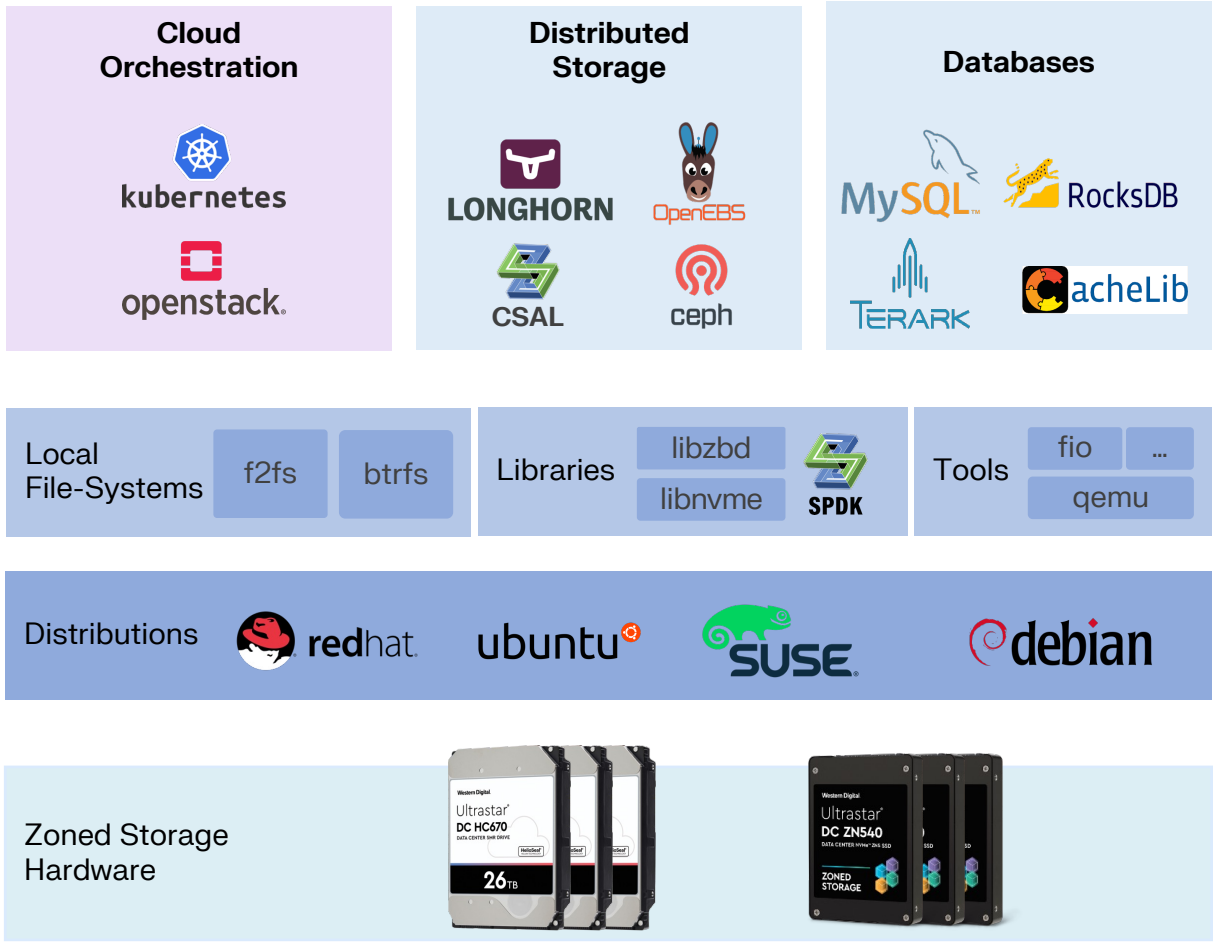
Zoned Block Device

Flash Memory Summit

→ Native Zoned Storage applications integrate with Zoned Cloud Native Storage solutions

Summary

- The Linux® zoned storage ecosystem continues to mature
  - BTRFS, Longhorn, Ceph, OpenEBS - Mayastor, CSAL, end-to-end zoned applications
- Workloads orchestrated by Kubernetes are now able to seamlessly benefit from Zoned Storage devices as cost-effective and high-performance storage
- **No longer software stack changes required to deploy and use zoned storage at scale**



Thank You!