OpenSDS Manageability using Swordfish for Cloud-native Frameworks

Anjaneya ‘Reddy’ Chagam, Sr. Principal Engineer, Intel Corporation
Steven Tan, VP & CTO Cloud Storage, Huawei
Notices & Disclaimers

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. **No product can be absolutely secure.**

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit [http://www.intel.com/benchmarks](http://www.intel.com/benchmarks).

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit [http://www.intel.com/benchmarks](http://www.intel.com/benchmarks).

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

The benchmark results may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user’s components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

© 2018 Intel Corporation.

Intel, the Intel logo, and Intel Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as property of others.*
Agenda

- OpenSDS Overview & Architecture
- OpenSDS Kubernetes Integration Architecture
- OpenSDS Swordfish Manageability
- OpenSDS Flash Management
- Summary
- Q&A
Overview

An open source community working to address data storage integration challenges, particularly in scale-out cloud native environments with heterogeneous storage platforms.
Goals

Open
- Standardized open software, services, and application framework
  - No vendor lock-in
  - Not under the control of a single or small group of companies
  - Accelerate development with shared components

Real
- Designed for real world use
  - Solve common end-user pain points
  - Optimize storage operations and utilization
  - Deploy in traditional or cloud-native environments

Ready
- Enable an ecosystem of ‘OpenSDS Ready’ suppliers
  - Mix and match ‘OpenSDS Ready’ hardware, software, UI, services, and apps
  - Hit the ground running with new projects
### Overview

<table>
<thead>
<tr>
<th>OpenSDS Platform</th>
<th>Control/Management Plane</th>
<th>Data Plane</th>
<th>Storage/Service Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard REST API, Single Pane Management, Policy-Based, Storage and Data Services, Orchestration and Automation</td>
<td>Data Reliability, High Availability, Data Protection, Data Mobility, Data Reduction, I/O Performance</td>
<td>DAS (HDD/Flash), SAN, NAS, AFA, Commodity, Private Cloud, Public Cloud</td>
</tr>
</tbody>
</table>

**Decoupled control plane from data plane and storage/service layer**
Manageability Focus

Orchestration & Automation
Data lifecycle, protection, replication, migration, security, governance, optimization

Multi-Cloud Data Management
Placement, access, security, and search across private and public clouds - AWS, Azure, GCP, and more

Data & Storage Intelligence
Data collection, monitor, reports, analytics, forecast, ML/AI
The Core Projects

**SUSHI**
The Northbound Plug-ins Project
Common plug-ins to enable OpenSDS storage services for cloud and application frameworks

**HOTPOT**
The Storage Controller Project
Single control for block, file, and object services across storage on premise and in clouds
OpenSDS Framework

OpenSDS

- Docker
- Kubernetes
- Mesos
- Cloud Foundry
- OpenStack
- OpenShift

Cloud Native Computing Foundation - CSI

OpenSDS Controller
Discover, Pool, Provision, Orchestrate

Cloud Integration

GCP
AWS
Azure
Others

Data Plane Components
- Direct Attached Storage
- Software-Defined Storage
- Enterprise Storage
- Cloud Storage

Extensions & Tools
- Deployment
- Monitoring
- CI/CD
- Service Mgt
- Networking
- Security
- BC/DR
- AI/ML
OpenSDS Architecture

Integrates with application/cloud framework orchestrators

Centralized, unified management and scheduling with REST API

Connects to storage backends for discovery, pooling, configuration, status. Allows driver integration, supports Cinder native drivers, and Ceph. Scale-out with storage.

- Written in golang
- Distributed etcd KV store for persistence
- Keystone authentication (& plans for others)
The Dashboard

Create Profile

Create profile to suit your specific use requirements.

- Name: high_performance
- Access Protocol: iSCSI
- Provisioning Type: Thin
- QoS Policy: Enable
  - MaxIOPS: 8000 IOPS/TB
  - MBWS: 100 MBWS/TB

Replication Policy: No
Snapshot Policy: No
Customization: Add
Available Storage Pool: Matching

OK  Cancel
Sample End User Scenarios

1. **Consolidation** of storage control and management across OpenStack and K8S clusters

2. **Snapshot backups** to cloud and restore to any storage platform

3. **Host-based replication** across different storage backends

4. **Data lifecycle management** to move data across storage tiers and archival for long term retention

5. **Multi-cloud data control** to allow the use of any cloud object store

6. **Orchestrate and automate** provisioning and backups across different storage platforms and backup software
OpenSDS Use Cases

1. Basic Operations (multi storage backend)

2. Array-based Replication

3. Host-based Replication

4. OpenStack Integration

5. Kubernetes Integration
Kubernetes Overview

- **Segment applications into microservices**
- **Package each part into its own container**
- **Dynamically orchestrate those containers to achieve optimal resource allocation**

**Master Server(s)**
- API Server
- Replication controller
- Scheduler

**Node**
- kubectl
- proxy
- kubelet

- pod
- Container

**Persistent Volumes**
- Scale Out
- Scale up (SAN, NAS, AFA)
Kubernetes Persistent Volumes

- A **PersistentVolume** (PV) is a piece of storage in the cluster that has been provisioned by an administrator.
- A PV can be provisioned *statically* or *dynamically*.
- A **PersistentVolumeClaim** (PVC) is a request for storage by a user through a StorageClass.
- A **StorageClass** provides a way for administrators to describe the “classes” of storage they offer. Different classes might map to different quality-of-service levels (or “profiles”) in other storage systems.
- A **StorageClass** needs to specify a provisioner for dynamic provisioning.
Container Storage Interface (CSI)

CSI is an industry standard defined to enable storage vendors to develop a plugin once and have it work across a number of container orchestration systems.

Source: https://github.com/kubernetes/community/blob/master/contributors/design-proposals/storage/container-storage-interface.md
OpenSDS Kubernetes Architecture

Kubernetes Master Server

Create PVC (Storage Class = Faster)

Node

Pod

OpenSDS CSI Plugin

OpenSDS Dock

Driver

OpenSDS Hotpot

Controller

Node

Pod

OpenSDS CSI Plugin

OpenSDS Dock

Driver

Vol

Storage

Gold Profile

Storage

Silver Profile

Storage

Archive Profile

OpenSDS provides driver integration and rich storage services (e.g. data protection, disaster recovery)
OpenSDS - Provision Volumes

Kubernetes Master

Controller Manager
PV Controller

API Server
Volume API

etcd
Snapshot CRD

Kubernetes Node

Pod for CSI node plugin

Kubelet
CSI Helper containers
Driver Registrar

OpenSDS.cs Plugin
NodeStageVolume
NodePublishVolume

Pod for CSI controller plugin

OpenSDS.cs Plugin
CreateVolume
ControllerPublishVolume
CreateSnapshot

CSI Helper containers
External Provisioner
External Attacher
External Snapshotter

OpenSDS Controller (Hotpot)

Cinder
3rd Party Storage
Cloud Storage Services (EVS, EBS, ...)

OpenSDS NBP (Sushi)

Dynamic Provisioner
FlexVolume Plugin
CSI Plugin
Service Broker
Other Plugins ....
OpenSDS – Profile Mapping to K8S

1. Create profile
2. Create StorageClass

Admin

OpenSDS
Kubernetes

StorageClass

3. Create PVC
4. PV Controller finds OpenSDS CSI Plugin
5. CSI Plugin asks OpenSDS to provision volume
6. OpenSDS chooses storage backend to create volume (PV)

User

Kubernetes

OpenSDS CSI Plugin

External Provisioner

OpenSDS

Cinder
3rd Party Storage
Cloud Storage

Cloud

PV Controller

Profile

Kubernetes

PV

Profile

Admin

OpenSDS
Kubernetes

StorageClass

3. Create PVC
4. PV Controller finds OpenSDS CSI Plugin
5. CSI Plugin asks OpenSDS to provision volume
6. OpenSDS chooses storage backend to create volume (PV)
Profile - Policy Driven Management

- OpenSDS profile is based on Swordfish specification.

- The SNIA Swordfish™ specification helps to provide a unified approach for the management of storage and servers in hyperscale and cloud infrastructure environments, supported by multiple storage vendors.

- An extension of the DMTF (Distributed Management Task Force) Redfish specification.
  - Redfish is designed by the DMTF’s Scalable Platforms Management Forum (SPMF) to create and publish an open industry standard specification and schema for management of scalable platform hardware. It is a RESTful interface over HTTPS in JSON format based on OData v4.

**Profile** is the key abstraction in “OpenSDS” for storage provisioning to support multi-cloud orchestration.
Profile - Swordfish Resource Mapping

Provisioning profile properties:
- DataStorageLoS
  - RecoveryTimeObjective
  - ProvisioningPolicy
  - IsSpaceEfficient
- I/O Connectivity LoS
  - AccessProtocol
  - MaxIOPs
  - MaxBWs

Snapshot profile properties:
- Schedule
  - Date
  - Time
  - Occurrence (daily/weekly/monthly)
- Retention
  - By number of snapshots
  - By duration to retain a snapshot

Data protection profile properties:
- DataProtectionLoS
  - Recovery Geographic Object
  - Recovery Point Objective
  - Recovery Time Objective
  - Replica Types
  - Consistency Enabled

Profile
- Provisioning profile properties
- Replication profile properties
- Snapshot policies
- Custom properties

Replication profile properties:
- DataProtectionLoS
  - Recovery Geographic Object
  - Recovery Point Objective
  - Recovery Time Objective
  - Replica Type
- Replica Infos
  - Replica Update Mode
  - Consistency Enabled
  - Replication Period
  - Replication Bandwidth
- Host Based Replication
  - Disk Drain (how to handle the ordering of dependent write requests)
  - Read Balancing
  - Resync Rate
  - Fencing (avoid split-brain)

Custom profile property examples:
- Disk Type
- Latency
- Deduplication
- Compression
- ......
Mapping Profile to Storage Systems

Profile
- Provisioning profile properties
- Replication profile properties
- Snapshot profile properties
- Custom properties

Selector

Capabilities
Storage backend 1
Capabilities
Storage backend 2
Capabilities
Storage backend 3

Capabilities
Storage backend 1
Capabilities
Storage backend 2
Capabilities
Storage backend 3
OpenSDS K8S CSI Plug-in

HighPerformanceSC.yaml

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: opensds-csi-high-performance-sc
provisioner: csi-opensdsplugin
parameters:
  profile: High-Performance

HighPerformancePVC.yaml

```yaml
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: opensds-csi-high-performance-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: opensds-csi-high-performance-sc
```

Note: profile parameter can be profile id or name
Running OpenSDS CSI Plugin

- Create OpenSDS CSI plugin pods:
  ```bash
  kubectl create -f csi/server/deploy/kubernetes
  ```
- Three pods can be found by `kubectl get pod`:
Using OpenSDS Volume

- Create nginx application
  ```bash
  kubectl create -f csi/server/examples/kubernetes/nginx.yaml
  ```

- An OpenSDS volume is mounted at `/var/lib/www/html`
  ```bash
  docker exec -it <nginx container id> /bin/bash
  ```

  ```bash
  root@nginx:/# mount | grep html
  /dev/sda on /var/lib/www/html type ext4 (rw,relatime,data=ordered)
  ```

```yaml
apiVersion: v1
type: Pod
metadata:
  name: nginx
spec:
  containers:
  - image: nginx
    imagePullPolicy: ...
csi-data-opensdsplugin
  persistentVolumeClaim:
    name: opensds-csi-high-performance-pvc
    readOnly: false
```
NVMe-oF: Local NVMe Performance

- The idea is to extend the efficiency of the local NVMe interface over a network fabric – Ethernet or IB
  - NVMe commands and data structures are transferred end to end
- Relies on RDMA for performance – Bypassing TCP/IP
- For more Information on NVMe over Fabrics (NVMe-oF)
  
**NVMe-oF in OpenSDS**

**Work In Progress**
- Linux Kernel Driver
- `nvme` commands for connect
- Drive assignment
- Profile includes target info

**2019 Plans**
- Multi-Target Pooling
- Rack aware scheduling
- Virtual Volume Provisioning
- NVMe over TCP/IP

---

Legend:
- Control Plane
- Data Plane
OpenSDS Roadmap v0.18

2017H2
ZEALAND
• Kubernetes
• Vol CRUD
• Standalone Cinder Integration
• CSI Support
• Ceph, LVM

2018H1
ARUBA
• OpenStack
• Replication
  Array-Based, Host-Based
• Dashboard
• Storage Profiles
• Enumeration
• Block Storage
  • Cinder Drivers
  • Ceph
  • LVM
  • Huawei: Dorado

2018H2
BALI
• S3 Object
• Multi-Cloud Data Control
• Monitoring
• Storage Groups
  • Snapshots, Replication
  • NVMeoF Preview

2019H1
CAPRI*
• Multi-OpenStack
• File Share
• Migration
• Data Protection
• NVMeoF

2019H2++
• Optimization
• Tiering
• Security
• Sharing
• Networking
Governance

Technical Steering Committee

- Steven Tan, Chair
  Huawei, VP & CTO Cloud Storage Solution
- Rakesh Jain, Vice-Chair
  IBM, Research Engineer and Architect
- Allen Samuels
  Western Digital, R&D Engineering Fellow
- Anjaneya “Reddy” Chagam
  Intel, Chief SDS Architect
- Jay Bryant
  Lenovo, Cloud Storage Lead
- Shinya Hamano
  Fujitsu, Software Architect

TSC
- Technical leads from vendor members
- Technical guidance and decisions
- Planning, roadmap

EUAC
- Technical leads from user members
- Requirements and requests
- Tests and feedback

End-User Advisory Committee

- Cosimo Rossetti
  Vodafone, Lead Storage Architect
- Yusuke Sato
  Yahoo Japan, Infrastructure Lead
- Kei Kusunoki
  NTT Communications, Storage Architect
- Yuji Yazawa
  Toyota ITC, Group Lead
- Wim Jacobs
  KPN, Senior Architect

Membership and Events  Project and Architecture  Community and Release  Ecosystem
Summary

- OpenSDS aim is to address *data storage integration challenges* in cloud native environments with heterogeneous storage platforms.

- OpenSDS is collaborating with SNIA to deliver Swordfish based DC manageability.

- OpenSDS community is fast growing - join and contribute to address emerging storage management challenges.
THANK YOU

https://www.opensds.io
https://github.com/opensds
info@opensds.io
@opensds_io
opensds.slack.com

FIND OUT MORE

Meet us at the OpenSDS booth!
Santa Clara, Sep 24-27

BE A MEMBER

Accepting New Members
Vendors And End Users Welcome
Backup Slides
Join OpenSDS

End User Benefits

1. Nominate a representative to the End-User Advisory Committee (bi-weekly online EUAC meetings)
2. Leverage OpenSDS technologies to transform storage for cloud native era
3. Seek help and support from OpenSDS TSC, EUAC and developer community
4. Propose projects, specify requirements, and vote on roadmap priorities
5. Participate in OpenSDS sponsored events; speak and exhibit
6. Get vendors to work together
Join OpenSDS

Vendor Benefits

1. Nominate a representative to the Technical Steering Committee (bi-weekly online TSC meetings)
2. Engage end-users and understand their needs first hand
3. Propose projects, specify requirements, and guide the roadmap
4. Participate in OpenSDS sponsored events; speak and exhibit
5. Collaborate and explore potential partnerships with other vendors in an open manner