OpenSDS-
An Industry Wide Collaboration For SDS Management

Lawrence Lai – Huawei
Rakesh Jain – IBM
State of Storage Management - Fragmented

Framework
- How to connect storage to a framework?
- How to provision storage for multiple frameworks?

Controller
- Which controller to use for the framework?
- Is my storage supported by the controller?
- Does it allow 3rd-party /IT integration?

Storage
- Is there a storage driver for the controller?
- Will I be able to leverage enterprise storage features?
- Can I use commodity hardware?
Storage Pains

Enterprise Customers
Want a common solution to manage storage needs for both internal and external customers
How about a single SDS controller to connect to all frameworks and storage backends?

Developers
Want a consistent method of consuming storage easily
How about a standard API that abstracts storage complexity?

Vendors
Want to enable their storage everywhere
How about standardization of storage controls that works with all frameworks?
Overview

OpenSDS is an open source community working to address storage integration challenges, particularly in scale-out cloud native environments with heterogeneous storage platforms.
Mission and Goals

The OpenSDS project promotes the use of simplified storage interfaces using a scalable storage controller architecture with open standard APIs, with the objective of providing application-oriented storage services.

- **Goals**
  - **Code**: To develop the open source core components and API’s needed for building an SDS solution
  - **Community**: To grow a community building value for end users and extends to other open-source communities
  - **Collaborate**: To collaborate among vendors, users, and standards bodies to solve real-world storage management problems
  - **Acceptance**: To get broad industry acceptance amongst users and vendors with the implementation and adoption of OpenSDS
Timeline

- Dell-EMC, Huawei, Hitachi, Intel
  - First Contact Mar 2016

- Linux Foundation Project Proposal
  - Project Proposal Sep 2016

- Hitachi, Huawei, Fujitsu, Western Digital, OSU, Vodafone
  - Project Launched Nov 2016

- Dell-EMC Joins Dec 2016
  - Prototype Demo Apr 2017

- CloudNativeCon Berlin

- IBM, NTT, Yahoo Japan
  - New Members! Jun 2017
Key Value Propositions—For End-users

**Cloud Transformation**
End-users can transform their existing storage into a platform for cloud native workloads

**Optimized Utilization**
Virtualization of on-demand storage resources, improving storage resource utilization

**No Vendor Lock-in**
Common integration and standardized API's keep storage ecosystem open
Key Value Propositions – For Developers

• Standard API offers a consistent method of consuming storage
• Abstraction of storage complexity enables developers to focus on their application design
• Develop once, run everywhere becomes a reality with a common storage control platform
• Accelerate application development without worrying about storage infrastructure
Key Value Propositions – For Vendors

- Unified SDS standards and development of open SDS solutions
- Efficient industry-wide collaboration and development
- Leverage OpenSDS ecosystem to build new solutions and businesses quickly
- Win-Win-Win for customers, partners, and company engineering and business
- Accelerate cloud transition leveraging the OpenSDS community
- Interoperability with OpenSDS solutions with standardization
- Reduce development cycle and improve efficiency so vendors can focus on real innovation instead of doing plumbing work
OpenSDS Project Framework

OpenSDS Plug-ins
- Cloud Native Computing Foundation - CSI
- Docker
- Kubernetes
- Mesos
- Cloud Foundry
- OpenStack
- Microsoft Azure Stack
- Others

Cloud Integration
- GCE
- AWS
- Azure
- HEC

OpenSDS Controller
- OPENSDS API
  - Block Services
  - File Services
  - Object Services

SOUTHBOUND DRIVERS AND PROTOCOLS
- Native Drivers
- Cinder
- Manila
- libStorage
- Swordfish
- Others

DATA PLANE COMPONENTS
- Local Storage
- SAN
- NAS
- NVMeF
- Others

EXTENSIONS & TOOLS
- Dashboard
- Monitoring
- Metering
- Performance
- Logging
- Backup
- DR
- Migration
- Tiering
- Forecast
- Analytics
- Others
Technical Introduction
Controller Project - The Universal Storage Controller

A universal storage controller with scalable architecture and open standard APIs

**Open Standard API**
Open API for block, file, and object services

**Policy-Based Orchestration**
Policy-based controls for provisioning, lifecycle management and more

**Wide Storage Support**
Comes with support for wide range of Cinder, and Manila storage drivers

**Enterprise Class Enabler**
OpenSDS API’s and policies enable the use of enterprise-class storage features
Plug-ins Project – Compute Framework Plug-Ins

A group of plug-ins enabling seamless integration with common compute and application frameworks

Common Standard Plug-in
Use a single common plug-in for a framework to access OpenSDS storage services

Seamless Integration
Plug-ins enable each framework to leverage OpenSDS storage services seamlessly

Simplify Storage
Abstract the underlying storage complexities away from the compute and application frameworks
Design Goals

Universal Storage Control
Single control plane for storage management

Abstract Storage Management
Discover – Available storage resources and their capabilities
Aggregate – Physical resources gathered into pools
Virtualize – Applications remain ignorant of storage physicality
Anonymize – Vendor neutral

Cloud Native Deployment and Scaling
Vendor-specific code confined to a single process
Support distributed and/or centralized server deployment model
Implicitly HA and scalable

Rules-based Centralized Controls
Simple rules map logical storage requests into physical storage requests
Design Requirements

Cluster-level View of Storage
Federate/aggregate multiple disparate storage systems into a seamless whole
Client and Target (Vendor) agnostic

Whole Storage Life-cycle Management
Full life-cycle management of storage (creation, access, destruction)

Universal Client Code
No vendor code in client

Zero Client Configuration

Common API
For all compute and orchestration clients
Expose and abstract enterprise storage functionalities
**OpenSDS Design**

**Common Interface**
Universal OpenSDS plug-ins for different frameworks. Open API for integration

**Policy-Based Controls**
Pooling of storage resources with policy-based controls for provisioning, data protection, lifecycle management and more

**Wide Storage Support**
Comes with support for range of Cinder and Manila storage drivers. Extensible for new storage

**Enterprise Class**
OpenSDS API’s and policies support x86 HW and enable the use of enterprise-class storage features
Advance Storage Controls

Advance API’s
OpenSDS provides API’s to drive policies for lifecycle, tiering, QoS, data protection etc.
Concept

- **Storage Profile**: A storage service profile created by the Admin – maps to one or more Virtual Pools

- **Service Catalog**: A list of Storage Profiles available for storage provisioning requests

- **Storage Request**: A storage provisioning request by a Tenant based on a Storage Profile

- **Admin**: Person that manages Storage Pools, creates Storage Profiles

- **Tenant**: Storage consumer that requests for storage through Storage Profile requests eg. End-user, application, K8S pod etc.
Concept

- **Region**: Each region has its own OpenSDS controller.

- **Zone**: Each region can be segmented into zones for fault isolation.

- **Physical Pool**: Storage resources can be grouped into physical storage pools eg. RAID group(s). Physical pool may span across nodes eg. Ceph pool.

- **Virtual Pool**: Physical pools of similar characteristics can be grouped together into a Virtual Pool across different Zones in a Region (future).

- **Storage Group**: A group of provisioned volumes/shares that allow operations such as snapshots to be performed together. Groups may be mapped to K8S pods, labels or services.
Storage Profile and Request

• Profile Name
• Regions {list of regions storage can be provisioned from}
• Protocols {list of data transfer protocols}
• Profile Policies (configured by administrator)
  • Access Control {list of tenants that can use this profile}
  • Max Request Size {max volume/share … size}
  • Security {data encryption options etc.}
  • Protection {data protection policy for this profile}
  • Replication {data replication policy for this profile}
  • Lifecycle (after detach) { retain | delete | erase }
  • Sharing { none | read write | read only } {list of tenants to share with}
  • Sharing {read-write, read-only}
  • Optimization {thin | compression | dedupe}
  • Custom

• Request Options:
  • Throughput
  • IOPS
OpenSDS Architecture Blocks

Cloud Orchestrator (OpenStack, K8S, Mesos, Swarm, etc.)

OpenSDS Plugin

OpenSDS API

OpenSDS Controller

Orchestration
(Lifecycle, Tiering, Data Protection)

Selector

Policy Engine

Metrics

Executor

OpenSDS Hub

OpenSDS Native

Other Drivers
OpenSDS for Cloud Native

Cloud-Native Storage
Integration with Kubernetes, Docker, and Mesos
- enables dynamic storage provisioning
- responds to container events eg. support for container migration to another host

Policy-Based Storage Controls
Built-in policies for lifecycle management, data protection, data security, and orchestrated controls for cloud-native apps

Cloud-Native Deployment
Connect to all storage backends supported by Cinder and Manila drivers

Built-in Support For OpenStack Storage
Connect to all storage backends supported by Cinder and Manila drivers

Storage Discovery and Pooling
Support discovery of storage backends and aggregation of storage resources into a seamless whole
OpenSDS for OpenStack

Policy-Based Storage Controls
Built-in policies for lifecycle management, data protection, data security, and orchestrated controls for cloud-native apps

Orchestrate Storage
Add orchestration to Cinder/Manila by automating operations such as snapshots, backups and lifecycle management

Leverage Enterprise Storage Features
Advanced OpenSDS API’s enables enterprise storage features to be fully utilized by OpenStack
Yahoo! Japan Introduction

- Yahoo! JAPAN is one of the largest internet company in Japan
- Our 100+ services earn 62billion PV per month
Yahoo! Japan: Infrastructure -1

- 6+ DCs (1DC in the US)
- 100,000+ Servers
- 50PB+ Storage
Yahoo! Japan: Infrastructure -2

- 50+ OpenStack Clusters
  50,000+ instances

- Next-gen Private Cloud Clusters
  Kubernetes, Cloud Foundry...
Yahoo! Japan: Storage Complexity

- Traditional Systems
  - NFS
  - HDFS

- Persistent Storage
  - Cinder
  - Manila
  - Swift

- OpenStack

- Kubernetes
- Cloud Foundry
Yahoo! Japan: Why We Join OpenSDS

- We need open, fair and long-term software project
- Reduce complexity
- Improve convenience
OpenSDS Demo

• **Scenario**: OLTP
• **Profile**: High Availability & High Performance
  ✓ **Device Type**: Huawei Dorado V3
  ✓ **Disk Type**: Dual-port NVMe SSD
  ✓ **Data Protection**: Active-Active DR cluster

• **Scenario**: Data Analysis
• **Requirement**: Large Capacity
  ✓ **Device Type**: IBM XIV
  ✓ **Disk Type**: SATA
  ✓ **Data Protection**: NA
For more information

Email: info@opensds.io
Web: www.opensds.io
Github: https://github.com(opensds)
Thank you!