Delivering a Standards Based SDS Framework with an Open Stack SDS Controller Implementation

Anjaneya “Reddy” Chagam
Principal Engineer
Intel Corporation
Agenda

- Data Explosion & Storage Pain Points
- Software Defined Storage (SDS) vision
- SDS Controller Use Cases
- SDS gaps and response
- Summary
Agenda

- Data Explosion & Storage Pain Points
- Software Defined Storage (SDS) vision
- SDS Controller Use Cases
- SDS gaps and response
- Summary
Data Explosion:

From 2013 to 2020, the digital universe will grow by a factor of 10, from 4.4 ZB to 44 ZB.

It more than doubles every two years.

Data needs are growing at a rate unsustainable with today’s infrastructure and labor costs.

...cost challenges continue to grow...

...data complexity is increasing

... and storage accounts for 40% of the data center budget.

Storage Pain Points

**Storage Silos (Traditional)**

- Application mapped to specific appliance
- Storage resources optimized to run specific workload
- Isolated storage resources

**Challenges**

- Cost of managing diverse storage solutions
  - Data Growth
  - Maintenance, Operations & Support
  - Infrastructure
- Vendor lock-in
  - Limited scalability
  - Flexibility to innovate
- Need for massively shared data
  - But not yet cloud ready

Traditional storage management is too complex and inefficient
Agenda

- Data Explosion & Storage Pain Points
- Software Defined Storage (SDS) vision
- SDS Controller Use Cases
- SDS gaps and response
- Summary
Software Defined Infrastructure (SDI)

**SERVICE ASSURANCE**
Policies and intelligent monitoring trigger dynamic provisioning and service assurance as applications are automatically deployed and maintained.

**PROVISIONING MANAGEMENT**
Orchestration provisions, manages and optimally allocates resources based on the unique requirements of an application.

**POOLED RESOURCES**
Network, Storage and Compute elements are abstracted into resource pools.

**Orchestration Software**
Orchestrates the delivery of services based on infrastructure attributes such as Power, Performance, Security, Thermo, and Utilization.

**Services Delivery**
Delivers services to applications based on their attributes.

**Infrastructure Attributes**
- Power
- Performance
- Security
- Thermo
- Utilization
- Location

**Application A**
- Application B
- Application C
- Application D

**Resource Pool**
- Storage
- Network
- Compute
SDS – A Key Component of SDI

Dynamic, policy-driven storage resource management

Services Delivery

Abstracting Software from Hardware, providing flexibility & scalability

Aggregating diverse provider solutions, increasing flexibility and drive down costs

Provisioning resources dynamically (pay-as-you-grow) increasing efficiency

Orchestrating application access to diverse storage systems through Service Level Agreements (SLAs), increasing flexibility and handle data complexity

SDS is a framework that delivers a scalable, cost-effective solution to serve the needs of tomorrow’s Data Center
**SDS Architecture**

**Data Services**
- Application that runs in data plane to optimize storage
- Ex: Predictive Analytics
- Ex: De-Duplication
- Ex: Tiering

**SDS Controller**
- Visibility and Control of ALL storage resources
- Communication between Apps, Orchestrator and Storage Systems
- Allocates storage resources to meet SLA’s

**Applications**

**Orchestrator**
- Visibility and control of ALL storage resources
- Communication between Apps, Orchestrator and Storage Systems
- Allocates storage resources to meet SLA’s

**Northbound API**
- Data Services
- SDS Controller
- SDN Controller
- Compute Controller

**Southbound API**
- Storage System [SAN]
- Storage System [Capacity]
- Storage System [Performance]
- Storage System [NAS]

**SDS : Consolidated Management of Scale-Out and Scale-Up Storage and plug into SDI**
Agenda

- Data Explosion & Storage Pain Points
- Software Defined Storage (SDS) vision
- SDS Controller Use Cases
- SDS gaps and response
- Summary
SDS Controller Use Case: Application Assignment (Logical Pools)

Example: Application provides storage pool

Gold
App 1 (OLTP)

Bronze
App 2 (Backup)

Silver
App 3 (Sharing)

SDS Controller

1. Discover
2. Compose
3. Assign

Gold
Storage System 1 [SAN]
High Perf IOPS – 10K (DC1)

Silver
Storage System [NAS]
Med Perf Tpt – 100Mbps (DC2)

Bronze
Storage System 3 [Distributed]
Capacity Tpt – 5 Mbps (DC1)

Gold
App 1 (OLTP)
High Perf IOPS – 10K (DC1)

Silver
App 2 (Backup)
Med Perf Tpt – 50 Mbps (DC1)

Bronze
App 3 (Sharing)
Med Perf Tpt – 5 Mbps (DC1)
SDS Controller Use Case: Application Assignment (SLOs)

Example: Application provides SLO attributes (performance)

<table>
<thead>
<tr>
<th>App 1</th>
<th>App 2</th>
<th>App 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IOPS=7000</td>
<td>Tpt – 2 Mbps</td>
<td>Tpt – 30Mbps</td>
</tr>
</tbody>
</table>

SDS Controller

Discover

Assign

1. Discover
2. Compose
3. Assign

- $$$ Gold
- $$ Silver
- $ Bronze

<table>
<thead>
<tr>
<th>SDS Controller Use Case: Application Assignment (SLOs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Perf IOPS – 10K (DC1)</td>
</tr>
<tr>
<td>High Perf IOPS – 5K (DC2)</td>
</tr>
<tr>
<td>Med Perf Tpt – 100Mbps (DC2)</td>
</tr>
<tr>
<td>Med Perf Tpt – 50 Mbps (DC1)</td>
</tr>
<tr>
<td>Capacity Tpt – 5 Mbps (DC1)</td>
</tr>
</tbody>
</table>
SDS Controller Use Case: Application Assignment (SLOs & Policies)

Example: Application provides SLO attributes (Throughput, noisy neighbor policy)

1. Discover
2. Compose
3. Assign

- **App 1**
  - Tpt – 2 Mbps

- **SDN**

- **SDS Controller**

- **Storage System 1** [SAN]
  - High Perf IOPS – 10K (DC1)
- **Storage System [Distributed]**
  - High Perf IOPS – 5K (DC2)
- **Storage System [NAS]**
  - Med Perf Tpt – 100Mbps (DC2)
- **Storage System [Distributed]**
  - Med Perf Tpt – 50 Mbps (DC1)
- **Storage System [Distributed]**
  - Capacity Tpt – 5 Mbps (DC1)

- **$$$ Gold**
- **$$ Silver**
- **$ Bronze**
SDS Controller Use Case: Storage Overflow

Example: Application provides SLO attributes (Performance but storage is full)

Policy #X
If 80% capacity
Then mark system unavailable

App 1 Tpt – 2 Mbps

1. Discover
2. Policies
3. Assign

SDS Controller

Free Pool

Policy #X
If 80% capacity
Then mark system unavailable

High Perf IOPS – 10K (DC1)
High Perf IOPS – 5K (DC2)
Med Perf Tpt – 100Mbps (DC2)
Med Perf Tpt – 50 Mbps (DC1)
Capacity Tpt – 5 Mbps (DC1)

81% utilization

### Storage System 1
- [SAN]
- High Perf IOPS – 10K (DC1)

### Storage System 2
- [Distributed]
- High Perf IOPS – 5K (DC2)

### Storage System 3
- [NAS]
- Med Perf Tpt – 100Mbps (DC2)

### Storage System 4
- [Distributed]
- Med Perf Tpt – 50 Mbps (DC1)

### Storage System 5
- [Distributed]
- Capacity Tpt – 5 Mbps (DC1)

### SDS Controller Use Case: Storage Overflow

Example: Application provides SLO attributes (Performance but storage is full)

Policy #X
If 80% capacity
Then mark system unavailable

App 1 Tpt – 2 Mbps

1. Discover
2. Policies
3. Assign

SDS Controller

Free Pool

Policy #X
If 80% capacity
Then mark system unavailable

High Perf IOPS – 10K (DC1)
High Perf IOPS – 5K (DC2)
Med Perf Tpt – 100Mbps (DC2)
Med Perf Tpt – 50 Mbps (DC1)
Capacity Tpt – 5 Mbps (DC1)

81% utilization

### Storage System 1
- [SAN]
- High Perf IOPS – 10K (DC1)

### Storage System 2
- [Distributed]
- High Perf IOPS – 5K (DC2)

### Storage System 3
- [NAS]
- Med Perf Tpt – 100Mbps (DC2)

### Storage System 4
- [Distributed]
- Med Perf Tpt – 50 Mbps (DC1)

### Storage System 5
- [Distributed]
- Capacity Tpt – 5 Mbps (DC1)
Agenda

- Data Explosion & Storage Pain Points
- Software Defined Storage (SDS) vision
- SDS Controller Use Cases
- SDS gaps and response
- Summary
SDS Gaps: Clear Standards

Standards exist, but they are not focused on SDS, not organized together, and are not complete.

Industry wide focus needed to develop standards by either enhancing or developing new standards (e.g., SNIA, OASIS, DMTF)
SDS Gaps: Standards Current State

<table>
<thead>
<tr>
<th>Standard</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>OASIS TOSCA</td>
<td>Applications provide storage requirements using SLOs</td>
</tr>
<tr>
<td>IETF</td>
<td>Network focus - storage requirements (e.g. policies) not comprehended</td>
</tr>
<tr>
<td>CDMI</td>
<td>Very good capability discovery for objects – need block/file extensions and granular capabilities (e.g. erasure codes).</td>
</tr>
<tr>
<td>SMI-S</td>
<td>Appliance focus - requires changes for distributed storage use</td>
</tr>
</tbody>
</table>
Standards: Application to Orchestration & Controller (SLOs)

**Wikipedia Definitions**

- **Service Level Agreement (SLA)** - A service-level agreement (SLA) is a part of a service contract where a service is formally defined. In practice, the term SLA is sometimes used to refer to the contracted delivery time (of the service or performance). SLA will typically have a technical definition in terms of mean time between failures (MTBF), mean time to repair or mean time to recovery (MTTR).

- **Service Level Objective (SLO)** - A service level objective (SLO) is a key element of a service level agreement (SLA) between a service provider and a customer. SLOs are agreed as a means of measuring the performance of the Service Provider and are outlined as a way of avoiding disputes between the two parties based on misunderstanding.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>SLOs (Examples)</th>
<th>Gold</th>
<th>Silver</th>
<th>Bronze</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>Latency (ms), Throughput (MB/s or IOPS)</td>
<td>&lt;10ms</td>
<td>&lt;20ms</td>
<td>&lt;30ms</td>
</tr>
<tr>
<td>Workload Profile</td>
<td>Sequential v/s random, Read v/s Write, Access pattern (burst, avg, min, max)</td>
<td>50:50 (R:W)</td>
<td>80:20 (R:W)</td>
<td>90:10 (R:W)</td>
</tr>
<tr>
<td>Cost</td>
<td>$/GB, $/IOPS</td>
<td>$2/GB</td>
<td>$1/GB</td>
<td>$.5/GB</td>
</tr>
<tr>
<td>Capacity</td>
<td>GB or TB</td>
<td>Max 2TB</td>
<td>Max 10TB</td>
<td>Max 2PB</td>
</tr>
<tr>
<td>Consistency</td>
<td>Eventual, Strong, Read after write</td>
<td>Strong</td>
<td>Eventual</td>
<td>Eventual</td>
</tr>
<tr>
<td>Durability</td>
<td>Replication, Geo protection</td>
<td>Local</td>
<td>Local</td>
<td>Geo</td>
</tr>
<tr>
<td>Availability</td>
<td>Up time (e.g., 99.999)</td>
<td>99.99</td>
<td>99.9999</td>
<td>99.99999</td>
</tr>
<tr>
<td>Compliance</td>
<td>HIPAA, ISO etc.</td>
<td>HIPAA</td>
<td>HIPAA</td>
<td>HIPAA</td>
</tr>
<tr>
<td>Protection</td>
<td>Encryption, Secure Erase</td>
<td>Client, Server</td>
<td>Client, Server</td>
<td>Server</td>
</tr>
</tbody>
</table>
Standards: Application to Orchestration & Controller (TOSCA, CDMI, ?)

Service Centric
Service Templates

- Need standard way of expressing application requirements using SLOs – work with OASIS, SNIA, IETF to create SLO standards

Storage Centric
e.g. Data System metadata

Source: 2011 SNIA Tutorial (Interoperable Cloud Storage with the CDMI Standard)

Source: https://wiki.openstack.org/w/images/a/a1/TOSCA_in_Heat_-_20130415.pdf
One policy framework is desirable in a data center for all resources (compute, network, storage). Need to incorporate storage requirements and policies in emerging standard discussions in IETF.
Standards: SDS Controller to Storage System

SDS Controller enables granular monitoring and QoS enforcement of storage data types (virtual volume, shares, containers) – need standards to address these gaps.
Clear Standards

**Open SDS Controller does not exist**

Open, federated “control plane” with pluggable architecture is needed for ecosystem innovation.
SDS Requirements

Provision (Storage System)
- Assign resource
- Provision resource

Discover (Storage Capabilities)
- Discover storage system, tiers
- Discover QoS
- Discover capabilities

Compose (Virtual Pools)
- CRUDL virtual pools
- Attach/Detach tiers to virtual pools

Data Type Life Cycle Mgmt
- CRUDL volumes
- Clone, Snapshot, Backup, Extend, Shrink volumes

- CRUDL shares
- Snapshot, Backup shares
- Attach object store
- Attach volume
- Detach volume
- Attach share
- Detach share
- Get volume stats
- Monitor QoS
- Alert Exceptions
- Data-lifecycle operations

Maintenance (Storage Systems, Pools)
- Deassign resource
- Unprovision resource
- Retire virtual pools

Note: Operations may be within storage system and among storage systems

Legend
CRUDL = Create, Update, Delete, List
## SDS Functional Partitioning

### Outside world (CLI, UI, Orchestration, Compute and Network Controllers etc.)

- **Provision** (Storage System)
- **Discover** (Storage Capabilities)
- **Compose** (Virtual Pools)
- **Maintenance** (Storage Systems, Pools)
- **Data Life Cycle Mgmt**

### SDS Controller

#### Block
- **Volume Operations** (Control Plane)
  - QoS Monitoring, Alerts
  - CRUDL volumes
  - Clone, Snapshot, Backup, Extend, Shrink, Migrate volumes
  - Attach volume
  - Detach volume
  - Get volume stats

#### File
- **Share Operations** (Control Plane)
  - QoS Monitoring, Alerts
  - CRUDL shares
  - Snapshot, Backup shares
  - Attach share
  - Detach share
  - Get share stats

#### Object
- **Object Operations** (Control Plane)
  - QoS Monitoring, Alerts
  - CRUDL = Create, Read, Update, Delete, List
  - Attach object store

### RAS system capabilities

- **Scheduling** (Storage resources)
- **Storage Operations** (among storage pools, tiers or systems)
- **Policy, SLO Mgmt**

---

**Legend**

- CRUDL = Create, Read, Update, Delete, List
SDS Controller Reference Architecture

**Orchestration**
- Users
- Service/Tenant Portal
- SDS Controller
- SDN Controller
- Compute Controller

**Control Plane**
- REST APIs
- SLA → SLO
- Provisioning
- Policy Mgmt
- Monitoring
- QoS

**Data Plane**
- SLA & App Reqs
- Storage Service
- Data Services (optional)

**Storage Systems**
- Distributed Storage
  - Object
  - SLOs
- Distributed Storage
  - File/Block
  - SLOs
- Storage Arrays
  - File/Block
  - SLOs

**Storage Hardware, Media**
- Unprovisioned

**Applications**
- Gold
  - Latency – 5ms
  - IOPS – 20000
  - Access - block
- Silver
  - Latency – 10ms
  - IOPS – 1000
  - Access - block
- Bronze
  - Latency – 50ms
  - Throughput – 100Mbps
  - Access - object

**SDS Controller**
- SLA → Storage SLO
- REST APIs

**MongoDB**
- eCommerce

**Media Store**
- Latency – 50ms
- Throughput – 100Mbps
- Access - object

2014 Storage Developer Conference. © Intel Corporation. All Rights Reserved.
Open Source SDS Controller - Prototype

1. SDS controller discovers storage systems and capabilities (e.g., perf, capacity, tiers, etc.)
2. Admin composes storage pools (e.g., gold, silver, bronze)
3. Application requests storage service using SLOs.
   Controller allocates storage volume from pool that can best service the request.
4. Storage gets assigned to Nova or App in VM.
   Controller works with compute, network to set QoS.

Develop key requirements by working with ecosystem and finalize open source enabling plans in 2H’14
SDS Gaps: Data Services

Clear Standards

SDS Controller

Data Services

Need a framework for vendor innovation

Data Services framework definition and standards needed for ecosystem innovation
Data Services are set of storage capabilities that manipulate data without much contextual interpretation.

Data Services can be deployed as virtual services or purpose built appliances.

SDS controller dynamically composes an end to end storage service by intelligently chaining data services and storage systems to meet application requirements. Can be extended to Network functions too (e.g. WAN acceleration for geo replication).

1. SDS controller discovers storage systems, capabilities
2. Application provides requirements using SLAs
3. Orchestrator provides SLOs to SDS controller
4. SDS controller composes storage resources and passes information to application
5. Application uses end point information for data path communication.

Data Services concept is similar to Service Chaining in Network Function Virtualization.
Agenda

- Data Explosion & Storage Pain Points
- Software Defined Storage (SDS) vision
- SDS Controller Use Cases
- SDS gaps and response
- Summary
Summary

• Software Defined Storage is needed to address tomorrow’s challenges
  - Data needs are growing at a rate unsustainable with today's infrastructure and labor costs
  - Traditional storage silos drives management complexity and inefficient
• SDS is a framework aimed at serving the needs of emerging storage requirements. But there are gaps -
  - Industry wide focus is needed to create standards for application and storage system interoperability
  - Open, federated SDS controller that provisions, monitors, provides SLA adherence is needed
  - Data Services framework definition and standards needed for ecosystem innovation
• SNIA has critical role in creating SDS framework and defining SDS storage standards by working with other standard bodies
Call to Action

• Contribute to Opensource SDS controller development

• Engage with SNIA to address SDS standards gaps
  - Develop cohesive Software Defined Storage Model with framework and standards in SNIA

• Pilot SDS solutions in 1H’15 (based on open source controller)

• Innovate storage offerings with Open source framework and Data Services
Backup Slides
What is TOSCA?

**Topology and Orchestration Specification for Cloud Applications**

A language for defining Service Templates...

...including a Topology Template describing the structure of a service

...including the definition of plans for orchestrating the application

Definition of building blocks for services

...along with the implementation artifacts for manageability operations

...and the definition of deployment artifacts for components

Definition of possible links between components

Cloud Service ARchive (CSAR)

Packaging format (CSAR) for packaging models and all related artifacts.

Source: https://wiki.openstack.org/w/images/a/a1/TOSCA_in_Heat_-_20130415.pdf
Using TOSCA to model applications

Source: https://wiki.openstack.org/w/images/a/a1/TOSCA_in_Heat_-_20130415.pdf
Using TOSCA for SLOs?

Requirements & Capabilities

- Some Application
  - Compute
  - Storage
- Database
  - arch = x86_64
  - cpus = 2
  - mem >= 4GB
  - OS = Linux
  - size >= 10GB

Database Provider
- provider = MySQL
  - version >= 5.5

Requirements can be fulfilled explicitly by other components in the model

Requirements can be fulfilled by the runtime

Requirements/Capabilities are base for substitutability

Provider A OR Provider B OR Provider C

Source: https://wiki.openstack.org/w/images/a/a1/TOSCA_in_Heat_-_20130415.pdf
SDS Controller Use Case: Application Assignment (Complex SLOs)

Example: Application provides SLO attributes (performance, client caching)

SDS Controller

1. Discover
2. Compose
3. Assign

App 1
IOPS=7000
Client Cache = 200GB

Storage System 1 [SAN]
High Perf IOPS – 10K (DC1)

Storage System [Distributed]
High Perf IOPS – 5K (DC2)

Storage System [NAS]
Med Perf Tpt – 100Mbps (DC2)

Storage System [Distributed]
Med Perf Tpt – 50 Mbps (DC1)

Storage System [Distributed]
Capacity Tpt – 5 Mbps (DC1)

---

Example: Application provides SLO attributes (performance, client caching)

App 1
IOPS=7000
Client Cache = 200GB

SDS Controller

1. Discover
2. Compose
3. Assign

### Tiered Services

- $$$ Gold
- $$ Silver
- $ Bronze

**Example:** Application provides SLO attributes (performance, client caching)

- **SDS Controller Use Case:** Application Assignment (Complex SLOs)
- **Example:** Application provides SLO attributes (performance, client caching)
SDS Controller Use Case: Application Assignment (Complex SLOs)

Example: Application provides SLO attributes (ephemeral)

- **App1**
  - SDS Controller
  - High Perf IOPS – 10K (DC1)
  - High Perf IOPS – 5K (DC2)
  - Med Perf Tpt – 100 Mbps (DC2)
  - Med Perf Tpt – 50 Mbps (DC1)
  - Capacity Tpt – 5 Mbps (DC1)

- **Ephemeral IOPS=170K (R/W)**

- **Discover**: $$$$ Gold
  - Storage System 1 [SAN]
    - High Perf IOPS – 10K (DC1)
- **Assign**: $$ Silver
  - Storage System [Distributed]
    - High Perf IOPS – 5K (DC2)
- **Compose**: $ Bronze
  - Storage System [NAS]
    - Med Perf Tpt – 100 Mbps (DC2)
  - Storage System [Distributed]
    - Med Perf Tpt – 50 Mbps (DC1)
  - Storage System [Distributed]
    - Capacity Tpt – 5 Mbps (DC1)
SDS Controller Use Case: Application Assignment (Complex SLOs)

Example: Application provides SLO attributes (ephemeral, local protection)

- **App 1**
  - Persistent, Local Protection
  - IOPS=170K (R/W)

**SDS Controller**

1. **Discover**
2. **Compose**
3. **Assign**

**SDS Controller Use Case: Application Assignment (Complex SLOs)**

- **App1**
  - SDS Controller
  - IOPS=170K (R/W)

**Storage System 1 [SAN]**
- High Perf IOPS – 10K (DC1)

**Storage System [Distributed]**
- High Perf IOPS – 5K (DC2)

**Storage System [NAS]**
- Med Perf Tpt – 100Mbps (DC2)

**Storage System [Distributed]**
- Med Perf Tpt – 50 Mbps (DC1)

**Storage System [Distributed]**
- Capacity Tpt – 5 Mbps (DC1)

**Examples:**
- **Gold**: Persistent, Local Protection
- **Silver**: Med Perf Tpt – 100Mbps (DC2)
- **Bronze**: Ed Perf Tpt – 50 Mbps (DC1)

**Example:** Application provides SLO attributes (ephemeral, local protection)

- **Example:** Application provides SLO attributes (ephemeral, local protection)
SDS Controller Use Case: Application Assignment (SLOs & Policies)

Example: Application provides SLO attributes (IOPS, noisy neighbor policy)

Storage System 1 [SAN]
- High Perf IOPS – 10K (DC1)

Storage System [Distributed]
- High Perf IOPS – 5K (DC2)

Storage System [NAS]
- Med Perf Tpt – 100Mbps (DC2)

Storage System [Distributed]
- Med Perf Tpt – 50 Mbps (DC1)

Storage System [Distributed]
- Capacity Tpt – 5 Mbps (DC1)

Qemu QoS
Max IOPS=9000

App1

Compute Controller

IOPS=9000

App1

Discover

Assign

Compose

IOPS=9000

Gold

Silver

Bronze

Example: Application provides SLO attributes (IOPS, noisy neighbor policy)

IOPS=9000

APP1

Qemu QoS
Max IOPS=9000

Compute Controller

Gold

Silver

Bronze

App 1

Discover

Assign

Compose

IOPS=9000

SDS Controller Use Case: Application Assignment (SLOs & Policies)