Scalable Storage Management with NVMe-oF

Piotr Wysocki, Rafał Bogdanowicz
Intel
Agenda

- Intel® RSD Overview
- NVMoF management in Intel® RSD
- Intel® RSD storage service evolution
- Intel® RSD 2.5 Redfish/Swordfish and SPDK mappings
Code availability

- All discussed code available on GitHub:
  - https://github.com/intel/intelRSD
  - It will be presented on SDC Workshop
Data Center Agility, Built on Open Standards

TODAY’S DATA CENTER CHALLENGES

Current Infrastructure

- Fixed ratio of compute, storage, and accelerator resources
- Expensive refresh & scale out
- Outdated software interface
- Cumbersome hardware provisioning process

“an industry-aligned architecture for composable, disaggregated infrastructure built on modern, open standards.”

Network
Server

Disaggregated
Composable
Interoperable

Increase Agility
Decrease Costs

2. Source: Disaggregated Server Architecture Drives Data Center Efficiency and Innovation, Shesha Krishnapura, Intel Fellow and Intel IT CTO, 2017
Intel® RSD Key Attributes

Disaggregated
- Network
- Compute
- Storage Sled
- Accelerator Sled

Buy less up front and Save money over time

Composable
- App 1
- App 2
- App 3
- Orchestration
- Intel Pod Manager
- Composed Node 1
- Composed Node 2

Compose hardware resources “on the fly”

Interoperable
- Single-Pane-of-Glass Management
  - Open Standard API
  - Vendor A Hardware
  - Vendor B Hardware
  - Vendor C Hardware
  - Vendor D Hardware

Choose the best now without vendor lock-in

OEMs with solutions based on RSD

Intel® Rack Scale Design

Intel® RSD Key Attributes
- Composable
- Interoperable
- Disaggregated

Operationalize IT resources on Intel® Rack Scale Design (RSD)

Compose hardware "on the fly" and choose the best now without vendor lock-in.

Intel® Rack Scale Design (RSD) provides a single-pane-of-glass management interface for easy orchestration and dynamic resource composition. It supports disaggregated, composable resources for flexible, cost-effective IT infrastructure.

Choose the best now without vendor lock-in, and buy less up front to save money over time.

Intel® Rack Scale Design is designed to integrate with a wide range of hardware vendors, ensuring interoperability and broad compatibility.
Benefits of Disaggregation and Composability

Disaggregated

- Network
- Compute
- Compute
- Compute
- Storage Sled
- Accelerator Sled

Buy less up front and Save money over time

Composable

- App 1
- App 2
- App 3

Orchestration

- Intel Pod Manager
  - Composed Node 1
  - Composed Node 2

Compose hardware resources “on the fly”

Resource pooling
Maximize utilization of high-value assets and improve agility with dynamic composability

Modular Refresh
Independently scale and upgrade resources with better lifecycle Management

Operational Costs
Improve Power Usage Effectiveness (PUE) and streamline operations and HW management

Resource pooling
- Maximize utilization of high-value assets and improve agility with dynamic composability

Operational Costs
- Improve Power Usage Effectiveness (PUE) and streamline operations and HW management

Modular Refresh
- Independently scale and upgrade resources with better lifecycle Management

Source: Disaggregated Server Architecture Drives Data Center Efficiency and Innovation. Shesha Krishnapura, Intel IT CTO, 2017

Source: Intel Pod Manager, Stanford University, Klimovic, Kozyrakis, et al, April 2016

**Intel® RSD – Composability**

Compose hardware resources “on the fly” for specific workloads

**Intel® RSD software functions include:**

**Resources Discovery**
Automatically discover and store hardware characteristics and location for all your resources

**Node Composition**
Dynamically compose compute, storage, and other resources to meet workload specific demands

**Telemetry Data**
Monitor data center efficiency and detect, diagnose, and help predict resource failures
Disaggregation

Save money over time with modular refresh

Great scalability

NVMe over Fabrics (Ethernet)
Intel® Rack Scale Software stack

**Cloud or Data Center Manager**
- VMware
- OpenStack
- 3rd Party Datacenter Mgr
- DIY

**Rack Scale Pod Manager**
- POD Wide Asset Management
- POD Wide Composed Nodes Management
- POD Wide Fabric Management
- POD Wide Storage Management

**Rack Scale Pooled Systems API**
- Pooled Systems
- Network Services
- Storage Services
- Rack Manager

**Pooled Systems**
- Compute Module
- Network Module
- FPGA Module
- Memory Module
- Accelerator Module
- Fabric Module
- Storage Module

**Network Services**
- Network Infrastructure
- Network Protocols
- Networking Services

**Storage Services**
- Storage Pools
- Logical Volumes
- Storage Endpoints

**Rack Manager**
- Power
- Thermal
- Chassis
Intel® Rack Scale Storage Services

Core Management Application

- Asset Manager

Redfish/Swordfish API

- CMDB

Storage Management

- Targets
- Logical Devices
- Physical Devices

Generic Assets Management Interface (JSON-RPC)

- NVMe over Fabrics
- iSCSI
- LVM Native Linux
- CEPH
- Native Linux Disk Mgmt

Operating System (Linux®)

Storage Assets (Physical and Logical)
Intel® RSD components in NVMeoF

POD Manager

Discovery Service

Storage Server

Client host (initiator)

Management network

Data network

Ethernet

Intel® RSD components in NVMeoF
Intel® RSD Storage Service evolution

Intel® RSD 1.2+
iSCSI Storage Service

- Redfish + OEM
  - PSME REST Server
  - GAMI
  - PSME iSCSI Agent
    - LVM
    - TGT-Admin

Intel® RSD 2.3+
Partitions over NVMe

- Redfish Swordfish
  - PSME REST Server
  - GAMI
  - PSME GPT Agent
    - GPT (parted)
    - Kernel (nvme)

Intel® RSD 2.4+
Storage Performance Development Kit

- Redfish Swordfish
  - PSME REST Server
  - GAMI
  - PSME SPDK Agent
    - JSON-RPC
    - SPDK (nvmf_tgt)
The Storage Performance Development Kit – a set of libraries providing:

- High performance
- Scalability
- Low latency
- Efficient use of CPU and memory resources
- Modularity
Storage Service and Common Fabric Model (Redfish + Swordfish)
Intel® RSD 2.5 NVM over Fabrics
SPDK to Redfish/Swordfish mapping

NVMe Block Device (BDEV) → /Chassis/{id}/Drives/{d-id}

Logical Volume Store (LVS) → /StorageServices/{id}/StoragePools/{sp-id}

Logical Volume (LVOL) → /StorageServices/{id}/Volumes/{v-id}

NVMf Subsystem → /Fabrics/{id}/Endpoints/{target-id}

NVMf Subsystem + Host → /Fabrics/{id}/Endpoints/{target-id}
→ /Fabrics/{id}/Endpoints/{initiator-id}
→ /Fabrics/{id}/Zones/{zone-id}
Initial discovery for all resources (GET)

- get_bdevs
- get_lvol_stores
- get_nvmf_subsystems
Intel® RSD 2.5 NVM over Fabrics
Redfish/Swordfish actions to SPDK mapping

Volume create (POST) → construct_lvol_bdev
Volume snapshot (POST) → snapshot_lvol_bdev
Volume clone (POST) → clone_lvol_bdev
Volume delete (DELETE) → destroy_lvol_bdev
Volume extend (PATCH) → resize_lvol_bdev
Intel® RSD 2.5 NVM over Fabrics
Redfish/Swordfish actions to SPDK mapping

- Endpoint create (POST)
- Endpoint delete (DELETE)
- Zone create (POST)
- Zone’s endpoints modification (PATCH)
- Zone delete (DELETE)

- `nvmf_subsystem_create`
- `nvmf_subsystem_add_listener`
- `nvmf_subsystem_add_ns`
- `delete_nvmf_subsystem`
- `nvmf_subsystem_add_host`
- `nvmf_subsystem_remove_host`
- `nvmf_subsystem_remove_host`
Questions?

Thank you