Next generation of ecosystem storage management

Balaji Marimuthu,
Manager Software Development, Microsemi
balaji.Marimuthu@Microsemi.com
Introduction

- With the current hyperscale datacenters, managing multi-vendor storage hardware using one simple user friendly tool is the datacenter admins desire. Server and storage Industries are trying to solve this common problem by providing a standard way of storage management. DMTF and SNIA have attempted to standardize the storage management using CIM and SMI-S standards for a decade. Now DMTF and SNIA have reviewed the lessons we learnt in a decade and have come up with Redfish and Swordfish. A simplified and easy to implement and use standards for the next generation of storage management. In addition to the standard based storage management, below are the common ask on the next generation of storage management.

- AI/ML based data analysis for prediction, notification and automated error recovery
- In-band and Out of band management
- Capability to run as containerized / server less application
Flavors of storage management tools

- GUI
- CLI
- APIs (SDKs, RESTful service,..)
- Plugins (vSphere WC, OpenStack Horizon, LSMCLI,..)
- Standalone scripts
- Microservices for a specific functionality
Storage management tools requirements

- Secure
- Standard based
- Easy to configure and provision the storage device
- Monitoring
- Notification
  - Alerts and events
  - Email and SMS notification
  - Push notification on mobile devices
- Firmware upgrade
- Progress task management
- Online help
- Discovery service
- Log management
- Scheduler service
- Statistics
- Topology view
- Fault detection and remediation
Storage management specification / protocol

- CIM / SMI-S
- Redfish / Swordfish
DMTF CIM (Common Information Model)

- The DMTF’s Common Information Model (CIM) provides a common definition of management information for systems, storages, networks, applications and services, and allows for vendor extensions.
- The CIM standard includes a Specification and a Schema, as well as a Metamodelf.
- The CIM Specification describes an object-oriented meta model. It defines the syntax and rules for describing managed objects in terms of meta schema elements.
- The CIM Schema provides the actual model descriptions.
- The CIM Metamodel defines the semantics for the construction of new conformant models and the schema that represents those models.
The Storage Management Initiative-Specification (SMI-S) is an ISO approved international standard that provides access to common storage management functions and features for the different storage vendors storage devices.

The SMI Architecture is based on Web-Based Enterprise Management (WBEM) from the Distributed Management Task Force (DMTF). The architecture is a client-server model that uses CIM-XML as the protocol. The client interface is the combination of the operations (e.g. get, modify, delete, enumerate, ...) defined in CIM-XML and the model defined in the Storage Management Initiative Specification (SMI-S).

https://www.snia.org/forums/smi/tech_programs/smis_home
DMTF Redfish

- The Redfish Scalable Platforms Management API ("Redfish") is a management standard using a data model representation inside of a hypermedia RESTful interface.
- Because it is based on REST, Redfish is easier to use and implement than many other solutions.
- Since it is model oriented, it is capable of expressing the relationships between components in modern systems as well as the semantics of the services and components within them.
- Uses Http(s) as web protocol and Odata-JSON for data format
The Swordfish Scalable Storage Management API ("Swordfish") defines a RESTful interface and a standardized data model to provide a scalable, customer-centric interface for managing storage and related data services. It extends the Redfish Scalable Platforms Management API Specification (DSP0266) from the DMTF.
Architectural implementation view

SMI Implementation

- SMI Client
- SMI Listener
- WBEM Server (CIMOM)
- SMI-S Providers
- Device(s)
- Instrumentation API

SMI Implementation

- Swordfish Client
- Swordfish Listener
- Redfish/Swordfish Server
- Swordfish Plugins
- Device(s)
- Instrumentation API

© 2018 Microsemi Corporation. Company Proprietary
Storage management communication paths

- **In band**
  - Traditional way for managing the storage products
  - Run on the operating systems
  - Standalone or a remotely manageable

- **Out Of Band / Side Band**
  - BMC based OOB is in use for more than a decade.
  - IPMI is used as a major communication protocol
  - Redfish is picking it’s share
  - All the new version of enterprise datacenter server are shipping with redfish based OOB
In-band Vs Out-Of-Band

Storage Management Apps

Operating system

BMC

Server Node 1

BMC

Storage Management Apps

Server Node 1
Storage management: products, server platforms, OS, browsers and language support

- **Languages**
  - English
  - German
  - Russian
  - Japanese
  - Chinese
  - Spanish

- **Browser Support**
  - Internet Explorer
  - Firefox
  - Chrome
  - Edge

- **OS Support**
  - Windows Server 2016
  - Windows Server 2012 R2
  - Windows Server 2012
  - Windows SBS 2011
  - Windows Server 2008 R2
  - Windows 10
  - Windows 8.1
  - Windows 7
  - Linux
  - Unix
  - Hypervisor

- **Server Platform**
  - x86
  - x64
  - PPC
  - ARM

- **Products**
  - SAS Controller
  - SAS Switch
  - SAS Expander
  - NVMe Controller

© 2018 Microsemi Corporation. Company Proprietary
Common challenges in In-band application deployment

- Even though we develop and validate the storage management application in multiple architecture and OS platforms, still the customer deployment environment vary a lot.
- Other vendor applications conflicting with the network ports, libraries versions and other shared resources.
Containerizing the application

Containers offer a logical packaging mechanism in which storage management applications can be abstracted from the environment in which they actually run. This decoupling allows container-based applications to be deployed easily and consistently, regardless of whether the target environment is a private data center, the public cloud, or even a admin/developer’s laptop.

**Consistent Environment**
Containers can include software dependencies needed by the application, such as specific versions of programming language runtimes and other software libraries. From the developer’s perspective, all this is guaranteed to be consistent no matter where the application is ultimately deployed.

**Run Anywhere**
Containers are able to run virtually anywhere, greatly easing development and deployment: on Linux, Windows, and Mac operating systems; on virtual machines or bare metal; on a developer’s machine or in data centers on-premises; and of course, in the public cloud.

**Isolation**
Containers virtualize CPU, memory, storage, and network resources at the OS-level, providing developers with a sandboxed view of the OS logically isolated from other applications.
AI/ML in storage management

Analysis:
• Analysis of the error reports
• Analysis of the IO stats
• Analysis of the alerts and events history

Prediction:
• Prediction of the failure based on the data analyzed
• Prediction of the possible options to improve the performance

Notification:
• Notification of the predicted failures and options to improve the performance

Remediation:
• Correcting/fixing the predicted failures
• Making the configuration changes to improve the performance
Storage management ML architecture

- Event logs
- Error logs
- IO stats logs

Event Normalization Module
Error Normalization Module
Stats Normalization Module

ML Algorithm

Trained data

Prediction report

Display/Notify to user

Remediation

Automated

Controlled

Config
Logical device failure prediction and correction

Input for ML
- IO errors
- Errors in the hard drive

Corrective actions:
- Run the consistency check based
- Replace the drive that is predicted to fail
- Add hot spare, in case of redundant logical
- Move the logical device to an another array of same class of service
Logical device performance enhancement

- Input for ML
  - Analysis of IO stats
  - Analysis of the configuration

- Corrective actions:
  List the options to improve the performance based on the IO stats and logical device configuration in comparison with the pre-defined trained data

  ✓ Fine tune the logical configuration to improve the performance.
    ✓ Stripe Size
    ✓ Queue Depth
    ✓ Cache
  ✓ Migrate the logical device to appropriate RAID levels based on the incoming IO type
  ✓ Migrate the logical device to the drives that are more suitable for the IO type
Hello, Welcome to maxView Chatbot. How can I help you?

I need to increase my logical-1 size by another 1TB in system1.

Let me check the system for possible options.

Logical-1 is currently 4TB. I find drives in slot 7 and slot 8 in enclosure 1 as options to increase the size. Which slot you want select?

Use Slot 8.

During expansion process Logical-1 will not be available to you. I hope you are aware of that.

Yes.

Initiated the expansion process. It would take time for the expansion to complete.

Do you need any further info/help?

No, That's all I need.

Great! Happy weekend and long vacation.
Q & A

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 4,000 employees globally. Learn more at www.microsemi.com

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer’s responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided “as is, where is” and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.