ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

Storage Management Initiative -Simulator

Anoop Pathak[1], BhirgeshSharma[2], Rahul Barwal [3], Vishesh Kumar Rathi [4], Yuvraj Gholap[5]

Student in Department of Information Technology, Army Institute of Technology, Pune, India[1,2,3,4]

Asst Professor, Department of Information Technology, Army Institute of Technology, Pune, India[5]

Abstract: An SMI-S provider is a vendor-specific module that is used as independent management software. An SMI-S provider is responsible for the actual processing of CIM operations on managed resources. The SMI-S Provider translates CIM-formatted requests into resource-specific operations and resource-specific operations to CIM-formatted requests. The SMI-S provider provides the mapping between the CIM interface and the resource-specific interface and contains the implementation for a set of CIM operations for a defined set of managed resources.

Keywords: SMI-S, CIM, WBEM, CIMCLI, SAN, CIMOM, CIM-XML etc

I. INTRODUCTION

To easily improve storage service levels and cost efficiency, most sites would like to use an integrated solution for managing storage performance. Except for rare cases, there is no longer a significant technical barrier preventing the adoption of a best of breed storage performance management solution such as the IntelliMagic Suite. The reason for this new flexibility is the market acceptance and maturity of the vendor neutral Storage Management Initiative Specification (SMI-S). The need for cross-vendor management tools has driven the adoption of this specification. Initial adoption of the specification by hardware vendors was lukewarm and implementations were immature. Fortunately for users, many of the hardware vendors are now very supportive of the SMI-S standards. Based on the current trend, it is likely that most new storage hardware platforms will include native SMI-S support.

This paper aims to suggest a Management Model of a storage system in distributed computing environment. Based on the Common Information Model and Web Based Enterprise Management, SMI-S is a standard to manage a storage system. This specification defines an interface for the management of a storage Area Network that is a heterogeneous environment of management applications, storage devices and storage system from different vender. Figure 1: Basic SMI-S Collection, provides a high level example of the SMI-S data collection flow.

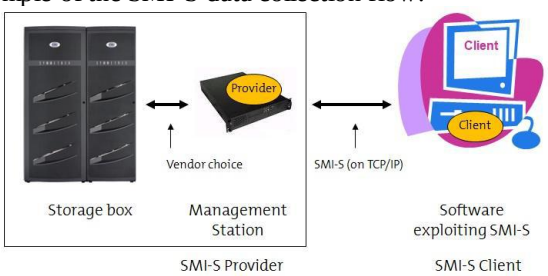

Fig 1: Basic SMI-S Collection

II. INSTALLATION AND COMPILATION

01) We need following RPM;
- tog-pegasus-libs-2.12.0-2.el6.x86_64.rpm
- net-snmp-libs-5.5-44.el6_4.2.x86_64.rpm
- lm_sensors-libs-3.1.1-17.el6.x86_64.rpm
- perl-5.10.1-131.el6_4.x86_64.rpm
- perl-libs-5.10.1-131.el6_4.x86_64.rpm
- openslp-2.0-0.2.beta2.el6.x86_64.rpm
- sblim-wbemcli-1.6.1-1.el6.x86_64.rpm
//wbem client

02) Install all the RPM.
rpm -ivh <package name>

03) Or just use 'yum' command.
yum install tog-pegasus
To install perl packages
yum provides */libperl.so
yum install perl

04) list down cimconfig file parameters
NOTE: cimserver must be active before setting the Cimconfig properties.
After configuration change, must be restarted.
Listing the config properties.
cimconfig -l -c
change the parameter values
cimconfig -s enableHttpConnection=true -p
cimconfig -s httpPort=5988 -p
cimconfig -s enableHttpsConnection=true -p
cimconfig -s httpsPort=5989 -p

05) Cimserver start/stop
/etc/init.d/tog-pegasus start/stop
OR # /sbin/service tog-pegasus start/stop
OR # service tog-pegasus start/stop

06) wbemcli command
wbemcli gc
'<http/https>://<username:passwd>@<localhost/ip>:<5988/89>/<namespace>:
<classname>

07) Access CIMObject remotely using wbemcli
- iptables rules may block the remote access of host.
To see iptables rules:
/sbin/iptables -L -n
Temporary clear all iptables rules;
/etc/init.d/iptables save
/etc/init.d/iptables stop

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

=>Pegasus installation using source tarball:
01) Download OpenPegasus source code tar file.
e.g. pegasus-2.11.2.tar.gz
Create directory for pegasus.
e.g. /opt/pegasus<version>
Extract the pegasus tar file to above location.
tar -xvf <filename>.tar.gz

02) Open bash_profile to set environment variables permanently.
vi ~/.bash_profile
export PEGASUS_ROOT= <path>/pegasus
export PEGASUS_HOME=$PEGASUS_ROOT
export PATH=$PATH:<path>/pegasus/bin
export PEGASUS_PLATFORM=LINUX_X86_64_GNU (use LINUX_IX86_GNU for CentOS)

03) Go to pegasus directory.
Compile the CIM server using 'make' command

04) Set the library path.
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PEGASUS_ROOT/lib

05) Go to pegasus directory and create repository.
make repository

06) Start/stop repository.
cimserver
cimserver -s

07) CIM server configuration
Specify userId to log on to the cimserver
cimuser -a -u <user_id>
Specify the authorization of this userId
cimauth -a -u <user_id> -n root/cimv2
Check the user authorization
cimauth -l

08) cimcli command to access cimobject locally
cimcli <cimoperation> -n <namespace> classname

09) To access cimobject remotely, install cimbrowser – CIMNavigator.
Install CIMNAVIGATOR tool on Windows 8
- Check java version running on windows
java -version
- if installed, then download cimnavigator zip folder from the site
http://cimnavigator.com/
unzip the folder. Go to /bin and edit the batch file. Correct the JAVA_HOME path and execute the batch file. It needs root administrative privileges to run the tool first time

10) Clear the iptables rules.
/etc/init.d/iptables save
/etc/init.d/iptables stop

11) Run the CIMNavigator tool.
Go to 'Edit' tab and select 'Server Configuration'
Edit the information, select the Ipaddress, namespace, CIM server type and port.
Afetr this, CIMNavigator get connect with reomote machine and user can browse all the CIM classes.

III. SMI-S

Storage Management Interface Specification, SMI-S, was created by the Storage Networking Industry Association (SNIA) in conjunction with the Distributed Management Task Force (DMTF) to develop and standardize interoperable storage management technologies. SMI-S is a common, standards-based management specification that permits third party applications the ability to configure and manage a storage array. Using the "provider" (the actual software library), a management application doesn't require knowledge of the specific architecture or infrastructure requirements of the particular storage platform. In the SMI-S architecture, client applications communicate with Storage Management Interface Specification (SMI-S) providers, or Common Information Model (CIM) agents, to obtain performance and configuration information from storage area networking components such as systems, fabric, and host elements.
SMI-S providers can report about asset, alerts, and performance information, as well as facilitate storage provisioning activities. SMI-S also provides reporting for switch and tape libraries. Each vendor provides a unique provider that facilitates SMI-S based reporting and management for their device.

SMI-S providers can be implemented either as proxies to the devices or as embedded software within the actual storage platform. Most legacy storage platforms have implemented their SMI-S providers as proxies. The proxies are software libraries external from the storage platforms that accept SMI-S queries and commands, and translate them into vendor specific commands which they send to the storage platforms. As the name implies, the embedded SMI-S providers are included on the storage platforms and do not require the installation or maintenance of a separate software package to provide an SMI-S interface to the storage platform. The trend for the newer platforms is to embed the SMI-S providers within the storage system as evidenced by the latest IBM DS8000 platforms and the EMC V-Max platforms.

The Common Information Model (CIM) is a hierarchical, object oriented architecture that is used to describe the attributes of managed objects in a enterprise computing environment. For example, CIM can be used to describe the characteristics of a computer system. CIM is also used to depict the relationships between different managed objects. For example, CIM can be used to depict the relationship of disks that are connected to a computer system. CIM consists of a specification and a schema.

Web Based Enterprise Management (WBEM) is a set of standards based technologies that are used to provide a uniform mechanism for exchanging CIM information between Clients and WBEM Agents in an enterprise computing environment. The Distributed Management Task Force (DMTF) defines a set of WBEM Operations that allow a Client to retrieve CIM data and to request that operations be performed on CIM data by the WBEM Agent. These operations are defined by the DMTF in the CIM Operations over HTTP specification. SMI-S 1.1.0 is based upon version 1.2.0 of this specification.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

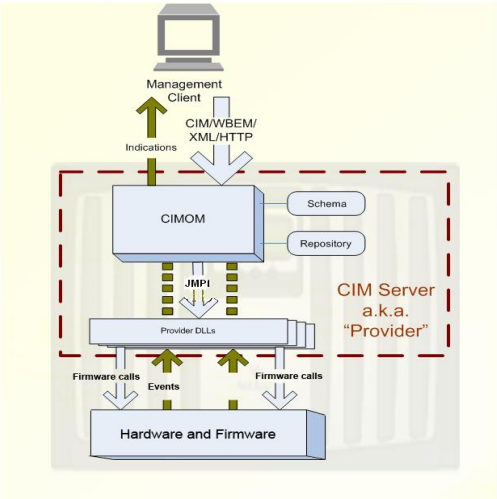

Fig 2: Provider Implementation

IV. PROFILES

A Profile defines the base set of information and capabilities that allow a Client to manage a particular storage resource such as a disk array. It defines the classes that a Client will use to perform a particular management task in a SAN. The Profile defines the associations that will be used to traverse between classes. In addition to identifying the class used, a Profile defines the properties and extrinsic methods of each class that must be supported A Profile can defines Subprofiles which represents additional capability that a vender can choose to make available. Like Profile, it defines the classes, properties and extrinsic methods that must be implemented to support its functionality. Also, it can incorporate Packages.

Likewise, a Package defines the classes, properties and extrinsic methods that must be implemented to support its functionality.
In SMI-S 1.1.0, the following groups of Profiles has been defined :

1. Storage : to manage different types of storage devices
2. Host : to manage components attached to host systems
3. Fabric : to manage the Fabric topology
4. Server : to manage the SMI Agent

Several Storage Profiles are defined for managing storage devices on a Storage Area Network.
Each Profile is focused on a different aspect of storage device management.

These devices are :
— Volume Management : allows a Client to manage physical disk as logical devices called volumes.
— NAS Head : allows a Client to manage a Network Attached Storage systems.
— Self-Contained NAS System : to manage a Client a Network Attached Storage systems.
— Storage Library : allows a Client to manage a storage system that has mechanism for retrieving data from different physical forms of storage media.
— Storage Virtualizer : allows a Client to manage a storage system that dose not directly include any local storage.

SubProfiles:
A Subprofile can be referenced by a Profile to allow optional inclusion of additional capability. A Subprofile defines the classes that a Client will use to perform the additional management tasks provided by the Subprofile.

Also it defines associations that will be used to traverse between classes. In addition to identifying the class used, a Subprofile defines the properties and extrinsic methods of each class that must be supported. However, a significant difference exists between a Profile and a Subprofile.

A Profile represents a base set of classes and capabilities that all supporting implementations must make available. In contrast, a Subprofile represents an optional set of classes and capabilities that a vender may or may not choose to implement. A Subprofile can contain the following components :

— The standards used
— The events that a Client can monitor
— The Packages that are incorporated into the Subprofile
— Etc

V. IMPLEMENTATION

We trying to develop CIMProvise, which is a tool that is aimed to act as a skeleton for fast provisioning of SMI-S capabilities in any storage array.

CIMProvise the Enabler

CIMProvise is designed with a plugin based framework in mind, where the top layer of providers will already be developed, and will have the capability to dynamically query any plugin for information as and when needed. If a storage array needs to grow its capability to include SMI-S control functionality, a CIMProvise plugin that talks respective storage CLI will be written and deployed with CIMProvise.

If that CLI is REST based, then a CIM to REST adapter needs to be written.

Information from multiple plugins can also be merged.

CIMProvise the Emulator

There are many applications that interact with storage using the CIM control path. For example, Storage Management GUIs, Snapshot management applications, Storage traffic analyzers. CIMProvise in its emulator form can be utilized to –

Quickly deploy an array with SMI-S control path capabilities and hence different test environments.
Inject errors at CIM layer.

Reduce resource contention on actual SMI-S capable hardware, as CIMProvise can be deployed as a virtual appliance.

CIMProvise can be mixed with other emulator tools, like nDisks to provide emulated control and data paths to perform end to end testing.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

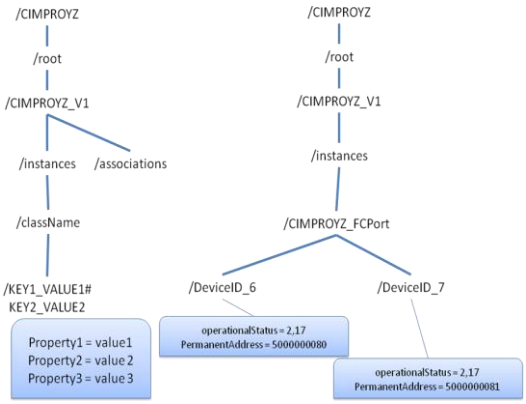

Fig 3:Hierarchy of CIMPROYZ

REFERENCES

[1] Storage Management Initiative Specification (SMI-S) v1.1, SNIA, October 2005.
[2] Web-Based Enterprise Management (WBEM), DMTF, 2003.
[3] Common Information Model (CIM) v2.10, DMTF, 2006.
[4] J. P. Thompson, "Web-based Enterprise Management Architecture", IEEE Communications Magazine, March 1998, pp. 80-86.
[5] J. G. Park et al., "A Method for Representing and Transporting CIM Operations Using Binary XML in the WBEM Architecture", ICACT 2006, Feb. 2006, pp. 1-4.
[6] J. P. Yang, "A Self-Configured WBEM Architecture to Upgrade the Storage Resource Management", COINNGNCON 2006, July 2006, pp. 120-122.
[7] OpenPegasus Release v2.5.
[8] CimNavigator v0.85, DuskFire Incorporation.
[9] Java Client SDK, WBEM Services v1.0.2, Sun, Nov. 2004.
[10] Common Manageability Programming Interface (CMPI) v1.3, Open Group, Sept. 2003.
[11] LVM HOWTO v0.19, AJ Lewis, Nov. 2006.
[12] The Software-RAID HOWTO v1.1, Emilio Bueso, June 2004.