Table of Contents

Information Provider Specification Proposal

Aim

The am of this proposal is to recommend a specification for information providers and to define the framework in which they are used. We require a specification so that information providers can be easily written by the many different developers of the systems we use. By having a specification we can ensure that all the information providers can be used in the same framework. A framework is also needed to manage the information providers. The different systems require information, eg local fabric management systems, grid information systems or grid monitoring systems should all be able to interface with the framework. This proposal advocates the use of existing open standards, in particular the standards from the The Distributed Management Task Force DMTF.

DMTF Interoperability

The DMTF is a standards body which his heavily backed by major industrial players. The Common Information Model (CIM) is the foundation for the DMTF technology solution to distributed enterprise management. The CIM schema is made up of two parts, the CIM meta schema and the CIM schema. The meta schema is a specification for describing schema. The CIM schema is a collection of defined schema which can be used for system management. The CIM schema can easily be extendended by defining vendor extensions and profiles are used to focus on a subset of the schema which is required for a particular task.

Web-Based Enterprise Management (WBEM) is a set of management and Internet standard technologies developed to unify the management of distributed computing environments. It includes protocols, query languages, discovery mechanisms, mappings, and anything else needed to exchange CIM information. CIM-XML is a WBEM protocol that uses XML over HTTP to exchange CIM information. The CIM Query Language (CQL) is heavily influenced by SQL.

WBEM implementations are based on a server, usually referred to as CIM Object Manager (CIMOM), running on the managed system and providing access to the system resources via the CIM Operations over HTTP protocol. The CIMOM handles request routing to object manager adapters, services (e.g. event services, query services, ...) and providers based on the CIM model.

Currently, the DMFT does not have a specification for the provider interface, however, The Open Group has defined the Common Management Programming Interface (CMPI) for information providers and this is now an industry accepted standard.

Open Source Implementations

There exists a number of open source projects that implement the DMFT standards. This following is just an example, it is not and exhaustive list!

CIMOMs

- openPegasus - Implementation from the open group.
- OpenWBEM - Enterprise-grade open source implementation.
- openCimom - Original SNIA open source Java implementation.
- WBEM Services - Open-source Java implementation.

Clients

- pywbem - Python WBEM client
- JSR48 - Java WBEM client
- sblim - Commad line WBEM client
- sblim - Java WBEM client
- rubywbem - Ruby WBEM client

Provider Interfaces.

- cimbiote - Python API
- cimple - C++ API
- sblim - Perl API
- owperl - Perl API

Open Pegasus and SBLIM packages are part of the SL4 distribution.

Open Pegasus

Open Pegasus is an open-source implementation of the DMTF WBEM standards. It is coded in C++ so that it effectively translates the object concepts of the CIM objects into a programming model but still retains the speed and efficiency of a compiled language. Pegasus is designed to be inherently portable and builds and runs today on most versions of UNIX, Linux, and Microsoft Windows. In addition, Open Pegasus is part of the Redhat Enterprise Linux Distribution.

Setting up Open Pegasus on SLC4

Use apt-get to install the rpm from the SCL4 distribution

```
apt-get install tog-pegasus
```

Start the service

```
/etc/rc.d/init.d/tog-pegasus start
```

Adding a provider to Open Pegasus

Open Pegasus requires two files; one MOF file which describes the output of the information the provider and the information provider itself. Open Pegasus uses C++ API for information providers as will as enabling use of the CMPI specification.

The MOF files are located in */usr/share/Pegasus/mof/* and the information providers are located in */usr/lib/Pegasus/providers/*

Testing Open Pegasus

Open Pegasus comes with an few example information providers. The */usr/bin/osinfo* command can be used to test the server and the default provider.

```
OperatingSystem Information
  Host: pcitgd10.dyndns.cern.ch
  Name: Scientific Linux CERN SLC
   ( 2.6.9-22.0.1.EL.cern #1 Thu Nov 3 20:19:44 CET 2005 )
  Version: 2.6.9-22.0.1.EL.cern
  UserLicense: Unlimited user license
  Number of Users: 2 users
  Number of Processes: 96 processes
  OSCapability: 32 bit
  LastBootTime: Nov 23, 2006  9:34:41 (00100)
  LocalDateTime: Nov 23, 2006  16:14:29 (00100)
  SystemUpTime: 23988 seconds =  6 hrs, 39 mins, 48 secs
```

Experimenting with the WBEM Interface.

The command *wbemexec* can be used to experiment with various XML requests. Create a file called *request.xml* with the following content.

```
<?xml version="1.0" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
 <MESSAGE ID="3075" PROTOCOLVERSION="1.0">
  <SIMPLEREQ>
   <IMETHODCALL NAME="EnumerateInstanceNames">
    <LOCALNAMESPACEPATH>
     <NAMESPACE NAME="root"/>
     <NAMESPACE NAME="cimv2"/>
    </LOCALNAMESPACEPATH>
    <IPARAMVALUE NAME="CLASSNAME">
     <CLASSNAME NAME="PG_OperatingSystem"/>
    </IPARAMVALUE>
   </IMETHODCALL>
  </SIMPLEREQ>
 </MESSAGE>
</CIM>
```

Running the command */usr/bin/wbemexec request.xml* will generate the following response.

```
<?xml version="1.0" encoding="utf-8"?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="3075" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateInstanceNames">
<IRETURNVALUE>
<INSTANCENAME CLASSNAME="PG_OperatingSystem">
<KEYBINDING NAME="CreationClassName">
<KEYVALUE VALUETYPE="string">
CIM_OperatingSystem
</KEYVALUE>
</KEYBINDING>
<KEYBINDING NAME="CSCreationClassName">
<KEYVALUE VALUETYPE="string">
CIM_UnitaryComputerSystem
</KEYVALUE>
</KEYBINDING>
<KEYBINDING NAME="CSName">
<KEYVALUE VALUETYPE="string">
pcitgd10.dyndns.cern.ch
</KEYVALUE>
</KEYBINDING>
<KEYBINDING NAME="Name">
<KEYVALUE VALUETYPE="string">
Scientific Linux CERN SLC
</KEYVALUE>
</KEYBINDING>
</INSTANCENAME>
</IRETURNVALUE>
</IMETHODRESPONSE>
</SIMPLERSP>
</MESSAGE>
</CIM>
```

Security

Open Pegasus has implemented SSL for http connections. Connections can use x509 based certificates for both server and client authentication.

SBLIM

The Interoperability Based Linux Instrumentation for Manageability SBLIM) is an IBM-initiated Open Source project, intended to enhance the manageability of GNU/Linux systems and it does so by enabling WBEM, Web Based Enterprise Management. The SBLIM project covers different aspects of an CIM enabled system. The available SBLIM packages can be subdivided into different categories.

1. Programming Interfaces
   - ♦ Allow a provider programmer to write CIMOM neutral instrumentation.
2. The CIM Instrumentation packages
   - ♦ Contain the CIM provider to access system resources.
3. CIM Client Applications
   - ♦ CIM client command line tool
4. CIM development environment
5. Development Tools section
   - ♦ Support to automate the creation of provider skeletons for certain interface types
   - ♦ A test suite to automate the function verification test of CIM instrumentation

Many SBLIM packages are part of the SL4 distribution

SBLIM WBEM CLI

The WBEM Command Line Interface is a standalone, convenient systems management utility for CIMOM access. Invocation and output syntax are problem-oriented and well suited for interactively inclusion in Shell and Perl scripts. It is especially suited for administrators writing their own management scripts or for WBEM developers that want to test their providers. The package can be installed using _ apt-get install sblim-wbemc_

The flowwing command would run the same information provider as used in the Pegasus web interface test.

```
wbemcli ei https://user:passwd@localhost/root/cimv2:PG_OperatingSystem
```

SBLIM CMPI provider

SBLIM also offers a number of existing information providers. The syslog information provider gives an example of how to use obtain information from a log file. The information provider can be installed using *apt-get install sblim-cmpi-syslog* Running the following command will dump the syslog file to stdout.

```
wbemcli ei https://user:passwd@localhost/root/cimv2:Syslog_LogRecord
```

SBLIM CMPI perl

The package *sblim-cmpi-perl* is an API which can be used for easily creating information providers in perl.

SBLIM CIM JDBC Mapper

With the CIM JDBC Mapper the CIM is transformed to a relational view so that all informations are presented in two-dimensional tables. The transformation is done in the following way:

- a CIMClass is a table
- an instance of a class is a row in the table
- the namespace identifies a database
- a class property is a column in the table

All operations are taking place on the client side. An SQL Statement is transformed in the semantic of the CIM Operations over HTTP protocol and sent to the connected CIMOM. Any given SQL Statement (must be supported) is processed afterwards and the results are passed to the classes of the JBDC Interface.

Comparisons to Glite

The CIMOM can be compared to an MDS GRIS. Both run information providers and present an interface which can be used to remotely request information. One of the main differences between MDS and a CIMOM is that the CIMOM analysis the request and only runs the information provider which are needed. The other differences is that the CIMOM only acts as an information broker and does not contains a cache.

In general MDS only returns the latest state of the system. As seen above with the syslog information provider, the WBEM framework can also provided logging or monitoring information. This is means that the WBEM framework could also be used in place of the current log file producers used in the Job Status and Grid FTP monitoring applications.

The WBEM clients would be a replacement for the LDAP APIs and ldapsearch command.

All the APIs for writing information providers can seen as providing the same functionality as the Generic Information Provider.

Summary

WBEM is an industrial backed standard for the management of distributed computing environments. There are a number of implementations of WBEM component and they are reaching maturing.

Advantages of using WBEM

1. Real Standard
    - ♦ Back by industry
2. Mature implementations exist
    - ♦ Reduce development and maintenance effort
    - ♦ Already in the RHEL distribution
        - ◊ Packages supported externally
3. Standard Interface for different monitoring systems
    - ♦ Can be simultaneously use by both Fabric and Grid systems
        - ◊ Many fabric management use the WBEM standards, the quattor team is considering them!
4. Standard Interface for information providers
    - ♦ Futher APIs available for simplicity
5. Can replace both GIP based "state" providers and log file style providers

Disadvantages of using WBEM

1. No caching or throttle control for providers
    - ♦ Could extend the existing provider interface to add this functionality
    - ♦ Become involved in the existing projects and suggest adding this feature.
2. Provider interface can be complex

- ♦ Use other projects which
- ♦ Create a generic provider for use within our project.
    - ◊ Use CIM/XML as the interface between scripts and providers.

Conclusion

Adopting WBEM technology would provide a number of standard interfaces which are required to solve a number of issues that we have. It is an industrial backed standard which has a number of implementations that are already part of the standard Linux distribution we use. The components would be a replacement for the MDS GRIS and provide site level transport for information and monitoring data. The framework could also be used to provide monitoring data for applications that we have such as GridView. Our current information providers limit the information to the "current state" data. The data could be used simultaneously by grid systems and fabric management systems. A number of APIs already exist to simplify the creation of information providers and vendors of various systems would be more likely to create information providers which follow an existing industry back standard. One of the major advantages is that only the provider needed to answer the request is run. This will make the system more efficient when dealing with dynamic data. One of the disadvantages is that there is no caching or throttle control for providers however, this could be solved by extending the information provider API to add this functionality.

This topic: EGEE > ProviderSpecification
Topic revision: r6 - 2008-11-17 - LaurenceField