Accelerated and Lightweight Web Management Model for Virtualization

Zhengang Li {lizg@cn.ibm.com} Chunyi Fu {cyfu@cn.ibm.com}*

Summary

Computer platform system virtualization is an evolving technology over the recent decades. Early as in 1972, IBM shipped the first operating system virtualization in the mainframe product line. To the recent booming investment from major industry players into the virtualization market, the scope of platform system virtualization spans over server, desktop market, and even some of the mobile device areas.

Earlier management of the virtualization was tied to the native virtualization core and was merely a clear layer in the whole architecture. The management interfaces were usually embedded or connected to the native host operating system. In this era, the management layer appeared as an accessory utility and never a separate layer and the interfaces for management were proprietary. With more and more players coming into the industry, open and universal management interfaces and models came into the market, e.g. the CIM-based SVPC model, as well as a couple of extended models encapsulating around this SVPC model. Though not based on a virtualization specific protocol, these models marked a milestone in standardizing the management model in virtualization technology. The management application and console need an additional installation on the user's computer system. Web technology is a lightweight layer suitable for most management scenarios. Both sharing the HTTP supporting protocol, the CIM and extended XML-based models are theoretically compatible with Web technology. Though most virtualization market players may have adopted these models more or less to improve the interoperability, usually a more comprehensive and rich featured proprietary management application or console is provided natively by the virtualization hypervisor using various interfaces. On the other side, third party contributed Web interfaces tend to be over-layered and are not dedicated to virtualization management, thus not focus on the management efficiency over existing virtualization in the Web architecture. There is an urgent requirement for a new model to fill in this area.

Using HTTP protocol to transfer the CIM-XML messages, CIM model requires a complex client side to interpret the messages and make responding message to the server. This placed a barrier between the SVPC model and the adoption to lightweight Web technology. We do see a lot of feature-rich AJAX frameworks that support a lot of complex Web browser side operations, but some of the CIM model's requirement requires too much browser side work around to implement, such as events pooling and serving for CIM indications. Even without the events support, encapsulating and decoding of the CIM messages pose a heavy work load for the browser, given that different browser implementations come with different XML decoding capabilities (originally for DOM object parsing). One of the extended models is to use WS-Man's WS-CIM standard to levitate the events pooling workload to the server side. This does simplify the browser-side events pooling work. But WS-Man uses SOAP messages, an even more complex XML formatted message, which is no easier for the browser to parse. On the other side of the HTTP protocol, the management server usually lives in the virtualization hosting system. Adding too much workload to the management server would compromise the hosting server system's capability to serve more virtualized guest workloads. We'll need a balance

* Zhengang Li & Chunyi Fu are with IBM China Systems and Technology Labs, 2F No. 10 Building, 399 Keyuan Road, Shanghai, China.

between the management workload and the real virtualization workload.

This paper presents a lightweight Web management model designed to accelerate the virtualization specific scenarios. Starting from the browser client, using RESTful protocol simply connects the two sides with shorter message body, where only variable and value pairs are passed along the channel. Following up on the server side, a well designed Web service gateway interface is serving the necessary property values. The properties come from serialized CIM instances that are going to be passed to the underlying lightweight CIMOM server, where an SVPC compliant CIM provider is working with the real virtualization layer. An independent events pooling component works between the gateway interfaces and the CIMOM server to do simple events collection and consuming. We fully deploy the SVPC model as part of the new model to remain compliant with the current virtualization players.

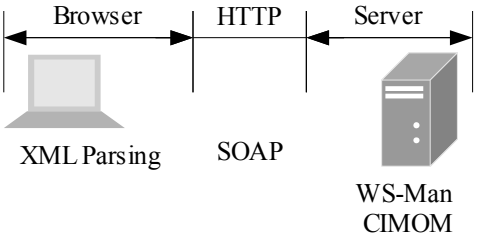

Figure 1: WS-Man/CIMOM Model

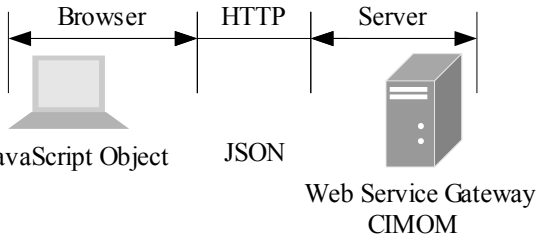

Figure 2: Lightweight Model

Original models suffer from the heavy workload of XML encapsulating and decoding, and a weighty layer of server hosting, such as WS-Man hosting. The presented model takes advantage of the RESTful protocol and a simple layer of Web service gateway to send only the necessary properties needed by the virtualization management. This brings the HTTP payload to a minimum level that doesn't require too much encapsulating and decoding on browser and server sides. Serialized CIM instance property values balance between compatibility and acceleration requirements.

We have implemented a 1:1 local console that manages self-virtualized servers by using this lightweight model (Figure 2). For the comparing purpose, a simple reference implementation is built using the WS-Man model (Figure 1) with WSCIM protocol over SVPC model. With the new lightweight model, users report that the browser side operations are more responsive. We also witness the server side memory footprint getting smaller.

The responsive user experiences come from less workload by directly using JavaScript object as values passed to the browsers. Without any delay in XML parsing, the interaction between the browser and the user is certainly better. The reduced server memory footprint comes from the lightweight layer of Web service gateway, which consumes fewer resources on the hosting server. It also gives more resources to the virtualization hosting.

A further step we would like to take in our research is to simplify the existing CIM/SVPC model to fit into this new model. This requires industry players to collaborate to keep it as neutral as a standard. Another direction is to directly connect the JavaScript object to the native virtualization layer to reduce even more footprint from the CIMOM server layer.