BMC Software Inc.

Technical Disclosure Publication Document

Authors

Itai Berman
Michael Theroux
Craig Sidorchuk

Integrated Operation of Distributed Enterprise Products through Unified Operations and Content Sharing

Posted: March 19, 2012

Overview

This proposal relates to enterprise products that are content-based and deployed over multiple sites. There are various reasons to opt for a multiple-site deployment; for example, it could be a means for achieving horizontal scaling or high availability. A real-life multiple-site deployment could result in a fairly complex system, with content distributed and developed in parallel on the different sites. This may pose substantial operational difficulties.

The following invention addresses these difficulties by introducing methods for creating and employing *Unified Operations* and *Unified Targets*. In addition, it shows how their deployment in conjunction with a content-sharing framework results in a synergistic process where site boundaries are removed and highly complex environments could be operated as a unity.

Background

When a product is deployed over multiple distributed sites, the distinct instances may turn out to be silos, forcing the end user to operate separately on each site. For example: a server automation product may be deployed over multiple sites, where each installed instance manages a subset of the server inventory. An operator wishing to execute a server automation process against the entire inventory may be forced to do that separately for each one of the underlying sites. Content management is another challenge in enabling the joint operation of an enterprise product. Each of the individual sites may be autonomous, having content and new workflows developed in parallel. Once content is

available in one of the sites, the operator may wish to use that against the entire deployment.

An additional complexity arises when product content is operated on targets. In this case, both the content and the associated targets may be segregated between different sites. Besides the additional distribution of entities, the operational problem becomes more accentuated as content may only be applicable to targets belonging to the same site. In other words: when the site identity is a fundamental piece in linking between content and applicable targets, it becomes even harder to provide a single unified operational interface that hides the underlying deployment complexities.

Solution

Unified Operations (UOs) are introduced in order to enable an integrated operation of an entire enterprise deployment. UOs capture abstract operations; each UO is mapped to one or more content pieces from the underlying sites for the actual implementation of the operation.

UOs address several use-cases in the realm of unified solution operation. At their core, they enable the execution of workflows on top of more atomic content pieces. A UO may be composed of a mixture of underlying content, possibly having some execution order applied on a type of execution, such as parallel versus sequential execution.

UOs enable the execution of operations, when applicable, across the solution's deployed sites, *without requiring the individual sites to be set explicitly*. This feature can be considered dynamic linking of content. By that, UOs become a powerful tool in the maintenance of product deployments with dynamic topology, as they enable automatic adjustment of the operational workflows to the changing deployment environment.

As an example, a UO could capture a process for analyzing the installed Windows server patches in a global IT environment, managed by multiple server automation deployments. The UO defines the process by linking it to content over the distributed sites. With the dynamics of an evolving deployment, where sites may be occasionally added or removed, and new servers are frequently added to the inventory, the UO enables the operation to be executed against the entire data-center at all times without the need to apply any change to it.

The full potential of UOs is arguably manifested when it is used in conjunction with a content-sharing framework. By using such a framework as an underlying mechanism, UOs are able to trigger and orchestrate the distribution of content based on logical relations, and to provide a unified operational interface that completely hides the inner intricacies of the distributed environment.

Driving content sharing through UOs is a natural extension of the underlying concept because UOs are designed to model logical entities spread across different sites. Thus,

they serve as appropriate means to push or synchronize content, either explicitly or behind the scenes (as a pre-step to the actual execution). Integrated with a content-sharing framework, UOs change dramatically the way users interact with a multi-site complex deployment. Content could be developed once and deployed anywhere; related content that is distributed over multiple sites could be treated as single entity, thus significantly reducing the problem of content profusion.

Unified Targets (UTs) are introduced in order to resolve the operational difficulties arising from the segregation of targets. UTs unify the targets spread across the different sites and products, similarly to the way UOs do so for operational content. The importance of UTs stems from the fact that in many cases the end-user would like to be completely agnostic to the distribution of targets across the sites. For example: if a product was deployed over multiple sites for the sake of horizontal scalability, such that each site controls a subset of the managed targets, the operator would typically want to have a unified view to the target inventory, regardless of the site distribution.
The combined usage of UTs along with UOs and a content sharing framework results in a powerful mechanism for unified operation of distributed, target-operated, multi-site solutions. In this case, UOs could be used to define global operations, while UTs would define the abstract targets to which the operations should be applied. By that, the user could operate while being completely agnostic to the underlying deployment. In case content is missing or not synchronized, the selected UT could drive a content-sharing process, ensuring the content is valid on all sites applicable to the targets.

Similarly to UOs, UTs could be evaluated across the product's deployed sites without requiring the individual sites to be set explicitly. In other words, they provide a mechanism to automatically map to targets across the inventory.

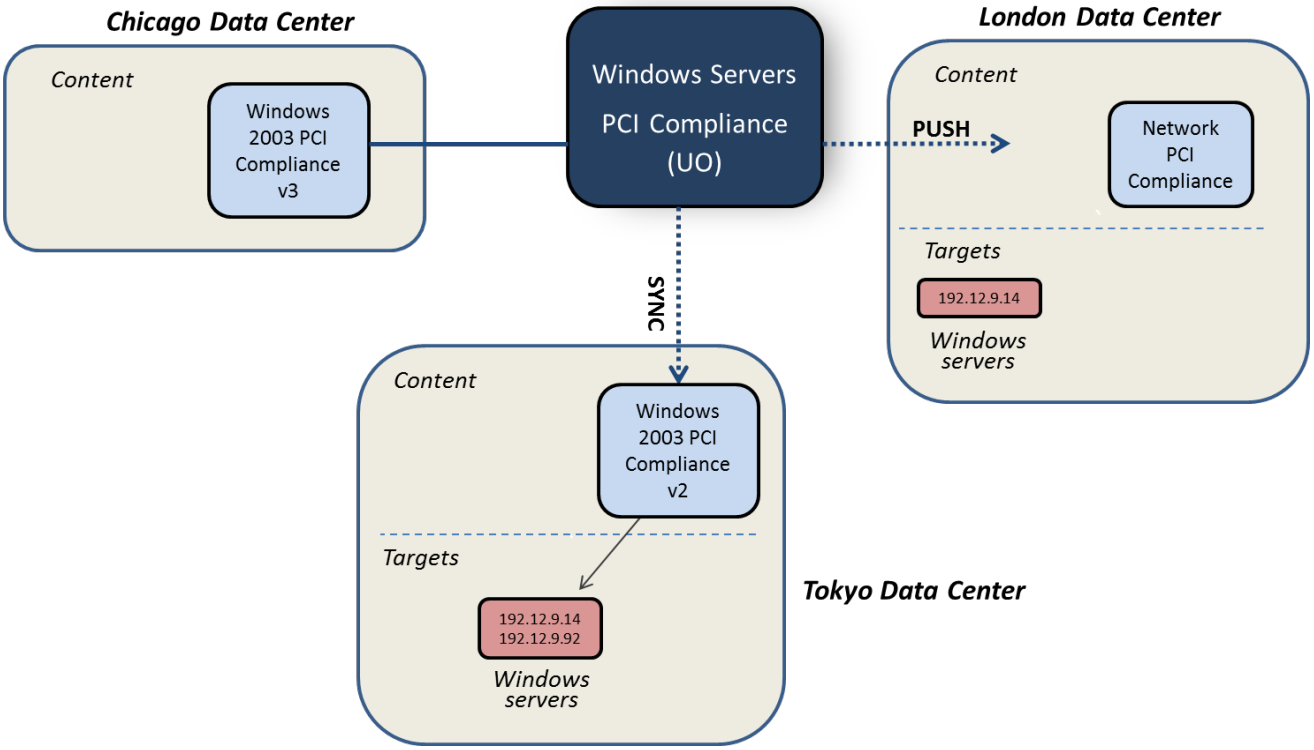
As an illustration, we could return to the previous example of analyzing the installed Windows server patches in a global IT environment, managed by multiple server automation deployments. A *Windows Servers* UT could be defined for further facilitating that operation. The *Windows Servers* UT will capture all the windows servers inventory managed by the different sites across the deployment, and be used as the target for the patch analysis operation.

Introducing UOs and UTs on top of a capable content-sharing framework enables a fairly complex deployment to appear as a single, homogenous environment. It turns the process of content distribution from an *automated* process into a completely *invisible* one, as site distribution and topology are no longer exposed to the end user. With that, multi-product/site deployments could be easily set up horizontally, turning the products almost infinitely scalable. Multi-tenancy and line of business separation could be achieved on products that do not natively support these concepts, in a manner that is invisible to the user.

Drawings

The following diagram illustrates how the usage of Unified Operations in conjunction with a content-sharing framework unifies site boundaries and data-center operation.

In this use-case, Windows 2003 PCI Compliance content is being developed in Chicago. The administrator would like to execute the latest version (v3) of the compliance content against two servers in Tokyo and one server in London. Through the dynamic linkage feature of UOs, it is identified that the Tokyo site does not have the latest version of the compliance content, and the London site does not have that content at all. Through the integration with content sharing, the UO execution drives the process of importing Windows 2003 PCI Compliance v3 to the London site, and synchronizing the Tokyo site with the latest version of that content. Now that the content is available on all required sites, the compliance checks may be executed. This example shows how this administrator could be completely agnostic to the complexities of site topology and distributed content development, enabling him or her to choose arbitrary content and target end-points when executing operations.



The following diagram further demonstrates the use of Unified Targets for enabling more complex use-cases in the distributed data-center operation.

Now, the administrator:

- Defines a 'Windows Server' UT, dynamically linked to the Windows Servers inventory in each individual site.
- Links the 'Windows Server PCI Compliance' UO to that UT.

With that, the administrator could easily execute a PCI compliance operation against the entire infrastructure. The master compliance content (located in Chicago) would be distributed across the infrastructure, and applied to all the servers managed by each site. This framework would automatically adjust any change in the content (e.g. a newer version of the compliance content) and any change in the server inventory (e.g. adding a new server in the Tokyo data-center). With that, the administrator is granted with a powerful method to scale his deployment horizontally once the inventory grows large, relying on the process described here to seamlessly adjust content distribution and operation execution.