DYNAMIC DATA CENTER AUTOMATION: An On-Demand Resource Provisioning Reference Architecture for the SAP® BusinessObjects™ Business Intelligence Platform

Version: 1.0
Date: October 2010

Editors: Chien-Hua Yen (Brocade), Kevin Liu (SAP)

Project Team:
Brocade: Chien-Hua Yen, Ezra Yosef, John Harcourt, Sharath Veldanda, Yasir Liaqatullah
SAP: Jay Thoden van Velzen, Siva Gopal Modadugula, Kevin Liu, David Cruickshank
VMware: Vas Mitra

BROCADE

vmware

SAP CO-INNOVATION LAB

Executive Summary

IT organizations are virtualizing the servers in their data centers because virtualization enables consolidation, and a more flexible server infrastructure. A virtual machine (VM) encapsulates all the software required to provide specific services. However, unlike traditional applications running on operating systems dedicated to specific server hardware, VMs decouple hardware from software so that services can run on any available hardware that provides a hypervisor to run the VM. And when demand on a VM-based service increases, additional VMs can be flexibly deployed onto any generic server in the data center, to be quickly assimilated into the virtual application pool for that service. This approach adds processing capacity and helps maintain service levels.

As usage of the SAP® BusinessObjects™ Business Intelligence (BI) platform increases, maintaining application performance while trying to meet varying levels of demand has become more challenging. In a virtualized data center or a private cloud, the SAP BusinessObjects BI platform shares, and often competes for system resources with many enterprise applications, while the demand on the SAP BusinessObjects BI platform shares can vary considerably. The traditional approach to addressing varying demand on the application is to reserve significant hardware resources, and over-provision the infrastructure to cater to anticipated demand spikes. However, that approach not only requires additional capital for hardware, as well as increased maintenance cost and energy consumption. In addition, the accurate prediction of system loading over time remains a challenge, and an over-provisioning strategy can frequently fail to meet changes in demand.

To maintain service-level agreements (SLAs), the system should instead adapt dynamically to load changes by providing accurate load monitoring and an on-demand resource provisioning service to the application. The SAP BusinessObjects BI platform, with its modular tiered architecture and support for virtualization, is an excellent candidate for such an on-demand resource provisioning service.

Brocade's ServerIron ADX with Application Resource Broker (ARB) automates on-demand resource provisioning with visibility into application traffic and load level. ARB connects to VMware vCenter through a VMware vSphere client plug-in, performing on-demand provisioning and de-provisioning of virtual machines (VM) based on performance metrics from the ServerIron ADX and VMware vCenter. If a load metric reaches a predefined threshold, ARB communicates with vCenter and with ServerIron ADX to initiate appropriate actions, such as powering on a virtual machine and adding the VM to the load-balancing pool. ARB provides application performance monitoring; dynamic, automated services to IT virtual infrastructure; optimized IT resource usage; and consistently sustainable SLAs.

To demonstrate the on-demand computing services provided by ARB, Brocade, SAP, and VMware have jointly defined a virtualized application reference architecture for the SAP BusinessObjects BI platform that closely models the customer deployment environment. A proof-of-concept (POC) testing of this reference architecture was performed at SAP Co-Innovation Lab in Palo Alto, California, using test workloads that closely resemble typical customer use cases.

The tests showed that ARB automatically provisioned SAP BusinessObjects BI platform VMs running in a VMware virtualization environment. The tests also revealed that ARB is simultaneously able to provision VMs across multiple tiers that may or may not include load-balanced traffic flows. The application deployment resulting from this solution adapts readily to load changes, maintaining customer SLAs while reducing the inefficiency and energy cost of over-provisioning to meet infrequent or intermittent demand.

This paper will first describe the on-demand computing architecture of the SAP BusinessObjects BI platform, software and hardware components, and configuration. Following the architectural discussion, two use cases are presented: Web tier on-demand VM provisioning, and Web tier and CMS tier on-demand VM provisioning.

Table of Contents

Overview

On-Demand Resource Provisioning for the SAP BusinessObjects BI Platform

The SAP BusinessObjects BI platform is an enterprise application that runs in multiple tiers. Each tier can benefit from virtualization by having one or more instances of similar server processes running on dedicated VMs to match changing business requirements. An important attribute of the tiered application architecture of the BI platform is that it can automatically detect any available services running in the environment, and dynamically distribute client load across them. This attribute is essential to drive an on-demand provisioning system. As a VM is brought online from either a suspended or a powered-off state, the BI platform services running on it will register with the BI platform cluster (CMS services) and be added into the BI platform service pool, ready to accept requests.

Despite the many benefits of a virtualized SAP BusinessObjects BI platform, maintaining SLAs and application performance remains a key area of concern for customers. System usage tends to change over time. More users running different types of reports may be a cyclical phenomenon – during the end of quarter, for example, responding well to planned remediation. Usage of the environment tends to change over time; for example, dashboards and analytical reports become more important at times than operational reporting. New users and departments coming online, or changes in business focus, can all lead to a change in BI needs. In a physical environment, such changes would require constant revisions of the system architecture and configuration. In the current, highly virtualized data center, the system administrator provisions VMs in response to demand changes. Even so, manual operations in a virtualized environment do not always provide optimal resource allocation, and often fail to precisely meet demand changes.

To simplify the management of a complex application infrastructure, the ideal system should intelligently and automatically adapt to changes in usage. The system should add VM resources during peak reporting cycles, for example, and then remove those resources when demand subsides outside those peak cycles. The highly virtualized data center that is running the SAP BusinessObjects BI platform needs an on-demand VM provisioning service that will complement the platform to provide an improved end-user experience. This on-demand provisioning service needs to have:

- Visibility into the infrastructure to measure system performance from server-to-client
- The ability to notify the system administrator to take remedial actions when needed
- The intelligence to automatically take appropriate actions based on predefined policies

ServerIron ADX with ARB[i] has the capabilities required to perform on-demand VM provisioning. As traffic flows through ADX, ARB retrieves configuration information, and application load metrics, such as application server response time, from ADX, and application server CPU usage from VMware vCenter. Its rules-based decision engine performs automated provisioning and de-provisioning of VMs based on these collected performance metrics. When a performance measure reaches its predefined threshold, ARB can either notify the administrator, or automatically allocate a VM from the resource pool. After the VM has started up, the OS Init script tells the BI platform services to share loads with other VMs of the same function.

ServerIron ADX with ARB provides application performance monitoring, optimized IT resource usage, and dynamic, automated services for virtual infrastructure. As such, it can be an essential part of the BI platform administrator’s toolkit, automatically and consistently helping maintain service-level agreements and customer satisfaction with the application.

On-Demand Provisioning Reference Architecture

The on-demand provisioning solution for the SAP BusinessObjects BI platform combines key components in computing, storage, networking, and software into a complete and integrated platform. The SAP BusinessObjects BI platform employs a multi-tier architecture (see Figure 1) to provide a flexible and scalable implementation of a virtualized environment. In the architecture, VMs can be added or removed to each tier to address changing demands. The Web/application tier contains multiple VM instances running the Tomcat application server (TC) and a load balancer to distribute user requests across the multiple servers. The Intelligence tier hosts Central Management Servers (CMS) to provide a service for connecting to the central repository database. The processing tier includes servers for SAP BusinessObjects Web Intelligence® software to process reports and dashboards.

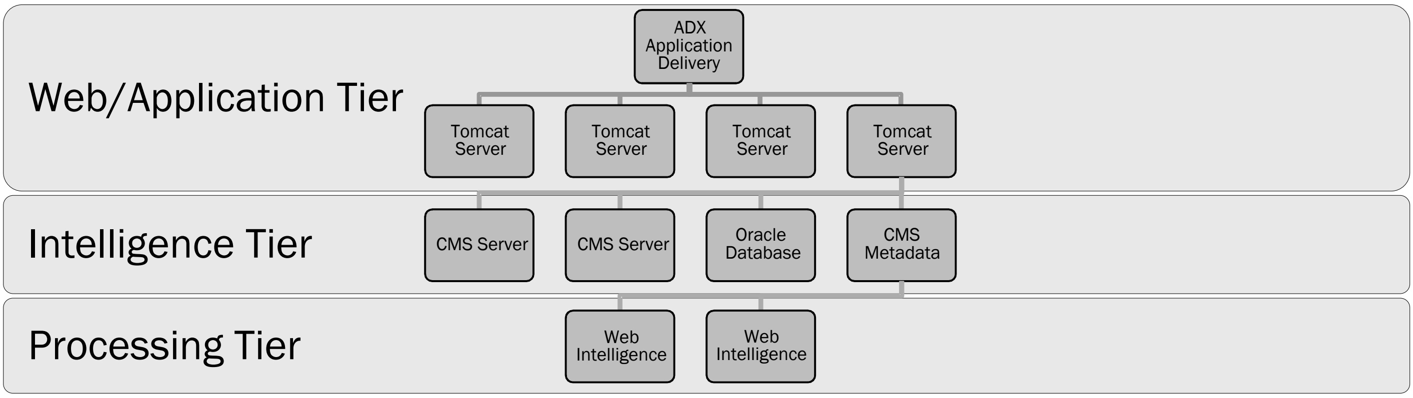

Figure 1 The SAP BusinessObjects BI Platform Logical Architecture

The Test Landscape

Major hardware components of the architecture include Intel Nehalem servers; the Brocade ServerIron ADX 1000 application delivery controller, the Brocade TurboIron TIX 24 10Gb Layer 2-3 switch, the Brocade 300 fiber channel switch, and NetApp Fibre Channel storage. Brocade provides end-to-end networking and fiber channel storage switching to the architecture. Major software components include SAP BusinessObjects BI platform 4.0, VMware vSphere 4.0, the VMware vCenter client, and Brocade Application Resource Broker. Figure 2 depicts the relationships between the hardware and software components.

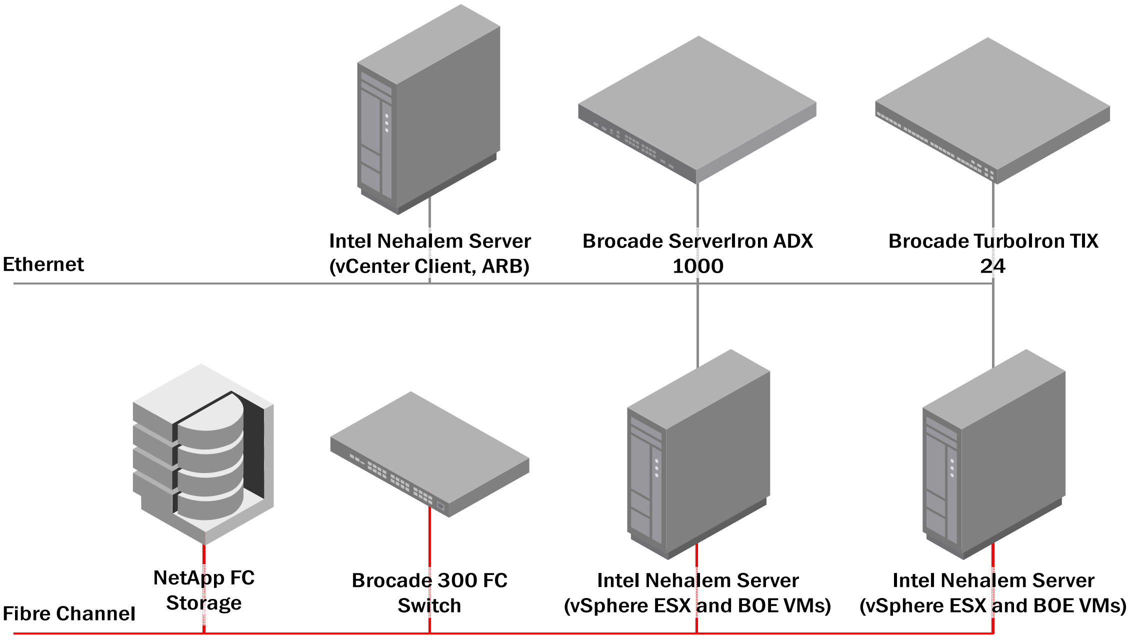

Figure 2 System Architecture

Table 1 lists the architectural hardware components described in this paper. Table 2 shows resource allocation for VMs of the SAP BusinessObjects BI platform.

Component	Product	Configuration
Server Hardware	2 – Intel Nehalem Server	Dual Socket, Quad core, 32 GB Memory VMware ESX, SAP BusinessObjects BI platform VMs
	1 – Intel Nehalem Server	Dual Socket, Quad core, 8 GB memory VMware vSphere Client, Brocade Application Resource Broker and ARB plug-in
Networking	Brocade ServerIron ADX 1000	16 Gb ports, L4-7 application delivery controller
	Brocade TurboIron TIX 24	24 Gb/10Gb ports, L2-3 switch
Storage	NetApp	Fibre Channel interface

Table 1 Hardware Components

Tier	Service	# VMs	Resource allocation
Web/Application	Tomcat Server	4	1 VCPU, 2G memory
Intelligence	CMS	2	2 VCPU, 4G memory
Processing	SAP BusinessObjects Web Intelligence	2	6 VCPU, 8G memory
Database	Oracle Database	1	4 VCPU, 8G memory
	CMS Metadata	1	2 VCPU, 4G memory

Table 2 System Resource Allocation

This hardware configuration supports approximately 400 concurrent users. An Intel Nehalem[ii] quad-core processor with Hyper-Threading technology provides eight hardware threads that can be scheduled by VMware ESX server. The total number of hardware threads available in the configuration, 32, is more than 26 VCPUs allocated for all VMs. While VMware ESX server allows for CPU over-commit, having sufficient hardware threads ensures minimal negative performance impact between VMs. This is an important consideration for an on-demand provisioning system in which provisioning rules are based on VM CPU utilization. If the CPU is overcommitted, even when CPU utilization is relatively low, overall system response may be slow because the VM is waiting to be scheduled to run. Sufficient memory is allocated to each VM to ensure that paging to storage does not happen during peak loading.

The Brocade ServerIron ADX 1000 application delivery controller provides Layer 4 through Layer 7 switching. ADX 1000 includes 16 Gigabit Ethernet (GbE) ports and two 10 GbE ports. It features predictive load-balancing functionality based on real-time application performance to efficiently distribute workload across multiple servers. In this deployment architecture, ADX 1000 provides a front end to the Web/application tier using a virtual IP (VIP) address. The VIP, which provides a single point of entry to access the application, masks clients from the servers and performs server load balancing, client connection management, and session persistence.

ServerIron ADX Configuration

ADX manages traffic using packet information beyond the traditional Layer 2 and 3 headers, connecting client requests to the most available servers based on the results of various Layer 4 and Layer 7 health checks. The switches efficiently distribute application services by measuring server connection load and response time, providing visibility and manageability of application performance, and service delivery.

For basic load balancing, the administrator needs to perform the following tasks on ADX:

- Define the real servers
- Define a virtual server (VIP)
- Bind the real servers to the VIP

However, a basic load-balancing setup needs to be optimized to run the SAP BusinessObjects BI platform. To use ADX for the SAP BusinessObjects BI platform without clustering the application server tier, the administrator needs to configure “sticky sessions,” where requests for a particular session are persistently routed to the same physical machine that first serviced the request for that session.

To meet this requirement, ADX uses advanced L7 content switching[iii] (CSW) to send requests with similar content to the same server. When a string in an incoming packet matches a rule, ADX uses a specified persistence method to select a destination server for the packet.

1. The CLI command for a rule that looks for packets containing a cookie header field with the string “ServerID=” would be:

```
csw-rule r1 header "cookie" search "ServerId="
csw-policy "p1"
    match "r1" persist offset 0 length 4 group-or-server-id
```

default forward 1
default rewrite insert-cookie "ServerID"

2. ADX examines packets to find a match with the persist string. The persist string is the portion of the matched content that ADX uses, along with the persist method, to calculate a destination real server or server group to which to send the packet. For example, for rule r1, defined above, we'll use the following for the content match.

   ServerID=1250

   The persist string could also be a segment of the matched content, starting from a specified offset and lasting for a specified length. In the example above, if you specify an offset of 0 and a length of 4, the persist string would be:

   1250

3. ADX uses the persist string along with the configured persist method to select a real server. For example, "*server-id 1250*" is assigned to the real server *vsvBOTC1 xxx.xxx.xxx.47*. All the port 8080 packets that contain the cookie "*ServerID=1250"* in their headers go to the real server *vsvBOTC1:*

   server real vsvBOTC2 xxx.xxx.xxx.48
   port 8080 server-id 1250

ServerIron ADX ARB Configuration

Before configuring ARB, the administrator has to define a baseline configuration for running the SAP BusinessObjects BI platform. A baseline configuration contains a set of VMs that do not belong to a VMware vCenter resource pool. When the load drops to zero and all VMs have been de-provisioned, the system is running in baseline configuration. The ServerIron ADX configuration should match the baseline.

The administrator also needs to create a VMware vCenter resource pool for each ARB application. An ARB application is one of the VIP-associated ports defined in the ServerIron ADX. A VIP may include many ports. Each VIP port is bound with a set of real server and port pairs. For example, a VIP in ADX is defined as:

```
server virtual demo.pal.coil xxx.xxx.xxx.109
port 8080
port 7800
port 9100
bind 8080 rsever1 8080 rserver2 8080      ! General HTTP traffic
bind 7800 rserve3 7800 rserver4 7800      ! Engineering HTTP traffic
bind 9100 rserver5 8080 rserver6 8080     ! Finance HTTP traffic
```

For this test ADX is configured to include three ARB applications: demo.pal.coil:8080, demo.pal.coil:7800, and demo.sap.coil:9100. Each ARB application has its own provisioning rule and can be provisioned independently.

ServerIron ADX with ARB uses performance metrics to provision VMs. Available performance metrics include application-server CPU usage, response time, and concurrent connections. The administrator configures rules to define threshold metrics per application. In order to write rules for on-demand provisioning, the administrator needs to first develop the baseline threshold of a rule by analyzing historical application data collected by ARB. After the threshold is established, the next step is to bind a resource pool with the ARB application. Administrators can create a set of rules for each ARB application. For example, the following rules were created for the Tomcat server running the SAP BusinessObjects BI platform:

1. Log an event when CPU usage is greater than 50%
2. Add a VM to the service pool when CPU usage is greater than 70%
3. Log an event when CPU usage is less than 20%
4. Remove a VM from the service pool when CPU usage is less than 10%

When ARB determines that a metric exceeds or has fallen below a threshold value over a predefined interval, it performs the associated action. After a provisioning event takes place, subsequent provisioning events will not occur until an adaptation period of four minutes has transpired following the provisioning event. This default adaptation period of four minutes can be modified in the ARB XML property file.

Use Cases

To validate the on-demand resource provisioning architecture as described in the previous section, SAP, VMware, and Brocade tested two use cases using HP LoadRunner to simulate loads of 250 users. The following sections present the test configuration, test scenario, and system CPU response during the test period.

On-Demand Resource Provisioning of the Web Tier in the SAP BusinessObjects BI Platform

The baseline configuration for this use case is shown in Table 3.

.

Tier	Service	# VMs	Resource allocation
Web/Application	Tomcat Server	1	1 VCPU, 2G memory
Intelligence	CMS	2	2 VCPU, 4G memory
Processing	SAP BusinessObjects Web Intelligence	2	6 VCPU, 8G memory
Database	Oracle database	1	4 VCPU, 8G memory
	CMS Metadata	1	2 VCPU, 4G memory
Resource Pool	Tomcat Server	3	1 VCPU, 2G memory

Table 3 System Resource Allocation

In this use case, the SAP BusinessObjects BI platform started with only one running Tomcat server VM, vsvBOTC1. CPU usage thresholds for adding and removing a Tomcat VM are set respectively at 60% and 40%. As loads increase, ARB adds a Tomcat server VM from Pool_1 that contains three Tomcat server VMs. Figure 3 shows CPU usage of vsvBOTC1, a Tomcat VM. CPU usage crossed the 60% threshold around 2:27 p.m., at which time ARB added a Tomcat server. CPU usage dropped immediately in response to the added VM. CPU usage started to increase again as HP LoadRunner added more users. ARB did not add another VM as CPU utilization crossed 60% again because the adaptation period had not expired. Around 2:34 p.m., after the adaptation period expired, ARB added another VM.

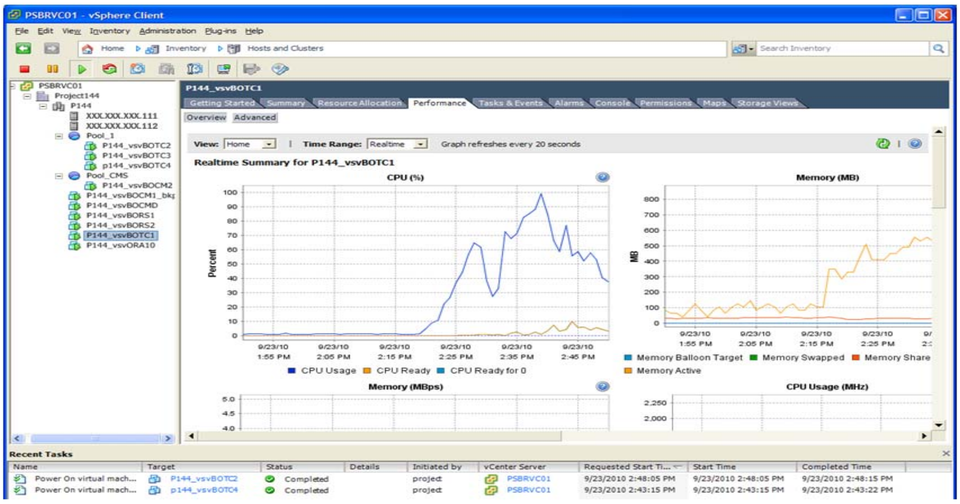

Figure 3 CPU Utilization for P144_vsvBOTC1

On-Demand Provisioning of the Web-Tier and the Intelligence Tier in the SAP BusinessObjects BI Platform

The baseline configuration for this use case is shown in Table 4 :

.

Tier	Service	# VMs	Resource allocation
Web/Application	Tomcat	1	1 VCPU, 2G memory
Intelligence	CMS	1	2 VCPU, 4G memory
Processing	SAP BusinessObjects Web Intelligence	2	6 VCPU, 8G memory
Database	Oracle database	1	4 VCPU, 8G memory
	CMS Metadata	1	2 VCPU, 4G memory
Resource Pool_1	Tomcat Server	3	1 VCPU, 2G memory
Resource Pool_CMS	CMS Server	1	2 VCPU, 4G memory

Table 4 System Resource Allocation

In this use case, the SAP BusinessObjects BI platform started with only one Tomcat VM and one CMS VM. CPU usage thresholds for adding and removing a Tomcat VM were set at 60% and 40% respectively. For adding and removing a CMS VM, they were set at 15% and 5% respectively. The purpose of this use case is to demonstrate that ARB can independently provision the Web and application tiers at the same time. The CPU threshold for the CMS tier is set artificially low to allow CMS provisioning to happen before Web tier provisioning. In a production environment, the CPU threshold for the CMS VM would be higher. Figure 4 shows CPU usage of vsvBOM1, a CMS VM. CPU usage crossed the 15% threshold around 2:27 p.m., at which time ARB added a CMS VM. CPU usage dropped immediately in response to the added VM. At about the same time, vsvBOTC1 also reached the 60% threshold, and ARB then added a Tomcat VM.

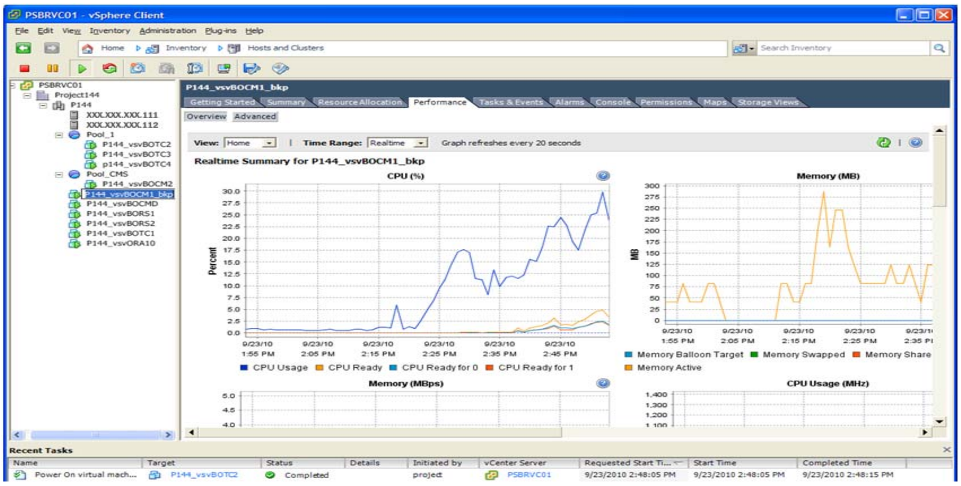

Figure 4 CPU Utilization for P144_vsvBOCM1_bkp

Conclusion and Recommendations

This paper presents an on-demand resource provisioning reference architecture for the SAP BusinessObjects BI platform running in a VMware virtualization environment. Using Brocade's ServerIron ADX with ARB, the architecture allows the SAP BusinessObjects BI platform to adapt to changes in demand and to maintain SLAs.

The system reacted to the changes in load being sent to the SAP BusinessObjects BI platform environment. As the load increased and crossed predefined thresholds, Tomcat-dedicated VMs were launched automatically in response, adding themselves to the environment. Similarly with increased demand for CMS and SAP BusinessObjects Web Intelligence services, as CPU thresholds on the running management (CMS) and reporting processing (SAP BusinessObjects Web Intelligence) tiers were crossed, indicating increased user activity, new VMs for each tier were launched by Brocade ADX with ARB, and upon launch these BI platform services added themselves to the larger BI platform environment. Similarly, as user activity decreased, each tier, based on low CPU thresholds, started shedding VMs to shrink the environment to its previous size.

The test shows that using advanced L7 context switching allows ServerIron ADX to meet the "sticky sessions" requirement of the SAP BusinessObjects BI platform in load balancing client traffic across all real servers. In each of the two use cases performed in SAP Co-Innovation Lab, ServerIron ADX with ARB automatically provisioned and de-provisioned BI platform VMs in accordance with predefined rules. The tests also showed that the ServerIron ARB could concurrently provision VMs in different tiers. Thus, the adaptive characteristics of this architecture apply to all tiers in the BI platform.

This paper establishes the groundwork for ARB enhancements and further optimization. Future work will include the SAP BusinessObjects Web Intelligence and SAP Crystal Reports tier, both of which provide CPU-intensive use cases. The workload will include multiple concurrent scenarios that resemble a typical customer's demands, and a deployment guide for the SAP BusinessObjects BI platform with ARB will provide configuration guidance for ADX and ARB.

Appendix

ServerIron ADX Configuration for Load Balancing

```
!Building configuration...
!Current configuration : 2172 bytes
!
ver 12.2.00B3T0000040
!
server source-ip xxx.xxx.xxx.60 255.255.255.0 0.0.0.0
!
context default
!
csw-rule "r1" header "cookie" search "ServerID="
!
csw-policy "p1"
 match "r1" persist offset 0 length 4 group-or-server-id
 default forward 1
 default rewrite insert-cookie "ServerID"
!
!
server real vsvBOTC1 xxx.xxx.xxx.47
 source-nat
port 8080
 port 8080 keepalive
 port 8080 server-id 1218
 port 8080 url "HEAD /"
!
server real vsvBOTC2 xxx.xxx.xxx.48
 source-nat
 port 8080
 port 8080 keepalive
 port 8080 server-id 1217
 port 8080 url "HEAD /"
!
server real vsvBOTC3 xxx.xxx.xxx.49
 source-nat
port 8080
 port 8080 keepalive
 port 8080 server-id 1216
 port 8080 url "HEAD /"
!
server real vsvBOTC4 xxx.xxx.xxx.50
 source-nat
port 8080
 port 8080 keepalive
 port 8080 server-id 1215
 port 8080 url "HEAD /"
!
server virtual vabadx01.pal.coil xxx.xxx.xxx.109
```

```
 predictor round-robin
 port 8080
 port 8080 csw-policy "p1"
 port 8080 csw
 bind 8080 vsvBOTC1 8080 vsvBOTC2 8080 vsvBOTC3 8080 vsvBOTC4 8080
!
vlan 1 name DEFAULT-VLAN by port
!
aaa authentication web-server default local
no enable aaa console
hostname COIL_ADX1
ip address xxx.xxx.xxx.105 255.255.255.0
ip default-gateway xxx.xxx.xxx.1
logging console
telnet server
username admin password .....
username root password .....
no-asm-block-till-bootup
!
end
```

ServerIron ADX Configuration for On-Demand Provisioning

```
!Building configuration...
!Current configuration : 2172 bytes
!
ver 12.2.00B3T0000040
!
server source-ip xxx.xxx.xxx.60 255.255.255.0 0.0.0.0
!
context default
!
csw-rule "r1" header "cookie" search "ServerID="
!
csw-policy "p1"
 match "r1" persist offset 0 length 4 group-or-server-id
 default forward 1
 default rewrite insert-cookie "ServerID"
!
!
server real vsvBOTC1 xxx.xxx.xxx.47
 source-nat
port 8080
 port 8080 keepalive
 port 8080 server-id 1218
 port 8080 url "HEAD /"
!
server virtual vabadx01.pal.coil xxx.xxx.xxx.109
 predictor round-robin
```

```
 port 8080
 port 8080 csw-policy "p1"
 port 8080 csw
 bind 8080 vsvBOTC1 8080
!
vlan 1 name DEFAULT-VLAN by port
!
aaa authentication Web-server default local
no enable aaa console
hostname COIL_ADX1
ip address xxx.xxx.xxx.105 255.255.255.0
ip default-gateway xxx.xxx.xxx.1
logging console
telnet server
username admin password .....
username root password .....
no-asm-block-till-bootup
!
end
```

References

[i] Intel Nehalem Architecture
[ii] Brocade ServerIron ADX Server Load Balancing Guide
[iii] Brocade ServerIron ADX ARB Admin guide