An Oracle White Paper
April, 2010

Effective Enterprise Java Virtualization with Oracle WebLogic Suite

“Oracle WebLogic Server on Oracle JRockit Virtual Edition means you don't have to make a tradeoff between the efficiency of virtualization and the performance of dedicated hardware. Oracle WebLogic Server on Oracle JRockit Virtual Edition running on Oracle VM delivers both -- along with unbelievable simplicity and security.”

- Adam Messinger, Vice President Oracle

Introduction

Increased operating costs, inefficient hardware utilization and rapidly expanding data centers have made virtualization the most compelling IT technology in years. Virtualization for the desktop and the server environment have evolved in the past years to finally deliver on its promise to lower operating costs and achieve better utilization of hardware. While virtualization has solved a multitude of problems, it is still difficult to deploy and operationalize complex application made up multiple tiers and components. Furthermore, virtualization is now moving into the commodity stage and the focus of virtualizing the operating system will change to be more directly targeted at virtualizing applications in order to reap the next level of benefits associated with virtualization. To accomplish this, organizations are evaluating and implementing Platform as a Service (PaaS) initiatives as a means of achieving greater cost efficiencies in the data center. Making this transition can be difficult as virtualization at this level has to date not been well supported by application and infrastructure providers. With the Sun acquisition under its belt, Oracle finds itself in a unique situation of being able to offer integrated capabilities throughout the entire technology stack that provide a complete integrated operationalized environment particularly well suited to offering PaaS, taking the promise of cost savings through virtualization to the next level.

Through innovations at the operational and runtime levels, Oracle WebLogic Suite provides an industry unique infrastructure enabling effective virtualization of Java EE deployments. This technical paper explores in detail the technology behind a new virtualized deployment paradigm for Java EE applications. Described below is not only the technology behind these advances, but also the benefits to the operational environment.

What is Virtualization?

Virtualization is the process of abstracting hardware resources, such as CPU, memory, storage, and network interfaces, such that multiple operating systems and applications can share the same hardware resources while maintaining strict isolation. The hardware runs virtualization software (e.g. a hypervisor) that enables the installation of multiple operating systems, each capable of running simultaneously and independently, in its own secure environment.

In a virtualized environment, each isolated partition runs an operating system and application. Each partition behaves as if it were a dedicated physical computer with its own [virtualized] CPU, network interfaces, storage, and operating system. Figure 1 shows multiple applications deployed in a hypervisor based virtualized environment.

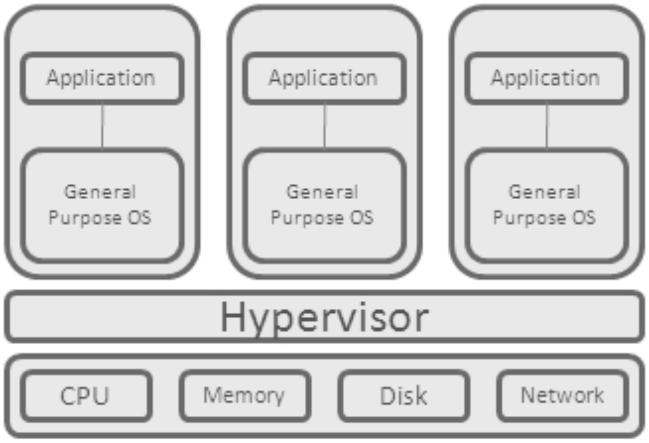


FIGURE 1: APPLICATIONS RUNNING IN A HYPERVISOR BASED VIRTUALIZED ENVIRONMENT

The goal(s) of virtualization for IT vary, but generally fall into one or more of the following categories:

1. Consolidate hardware to increase application density; save on cooling, rack space.
2. Increase utilization of provisioned hardware.
3. Accelerate provisioning of new applications and infrastructures.
4. Better manage dynamic fault tolerance and re-configuration of deployed systems.
5. Provide isolation of operating systems and applications on the same hardware.
6. The ability to run both legacy and new software on the same hardware.

Middleware Virtualization Challenges

While most middleware can function perfectly fine in a virtualized environment, application specific management challenges such as the ability to effectively manage and easily deploy complex applications materialize. Since most middleware applications are comprised of multiple tiers (UI, computational, caching, persistence etc.) the rollout of new deployments can be difficult as there are few facilities within traditional virtualization infrastructure to guarantee consistency and correctness of deployments. To make matters worse, migrations from different environments such as development, test and production are further complicated when dealing with virtualized environments. To address these concerns and challenges, WebLogic Suite adds additional capabilities to effectively manage virtualized deployments as well as provide for an optimized environment specifically targeted at Java EE applications.

Three key technologies help solve the operational complexity problem of deploying virtualized environments: Oracle VM, Oracle Virtual Assembly Builder and Oracle WebLogic Server on Oracle JRockit Virtual Edition.

Oracle Fusion Middleware, such as SOA Suite and WebCenter based applications, making use of WebLogic Suite for their runtime can take advantage of these new virtualization capabilities. All applications certified on WebLogic Suite are certified to execute on Oracle VM as well as use capabilities available through Oracle Virtual Assembly Builder. For SOA based applications, in particular, which tend to be complex, the notion of being able to create virtual appliances and form simple to deploy assemblies is compelling as this simplifies and accelerates deployment, reducing the risk of configuration errors as the application moves through the natural environment life cycle (development, test and production).

Moving from a physical to a virtual (P2V) environment is at first glance simple; create virtual images of the physical deployments. Inevitably, this results in image sprawl creating maintenance overhead as each of the images must be patched at the OS and application layers creating for unanticipated costs. Below, we will see how software assemblies address this issue through re-use of images across applications.

Java middleware is already virtualized through the Java virtual machine. Executing this environment on a hypervisor introduces undesirable application latencies. For custom Java EE applications, an optimized runtime environment is now available through Oracle WebLogic Server on Oracle JRockit Virtual Edition, allows for significantly improved performance over traditional virtualization environments.

Software Assemblies

An assembly is a collection of interrelated software appliances that are automatically configured upon deployment. Assemblies are typically deployed onto a set of virtualized hardware resources to ensure high levels of hardware utilization and efficiency.

While assemblies are simply a collection of virtual appliances with defined interconnects, assemblies must adhere to a set of base principles in order to be useful in a production environment; including:

1. Allow for the composition of virtual appliances as well as external systems.
2. Externalize configuration in the form of metadata that can easily be modified without potentially damaging the assembly itself. For example, virtual IP addresses virtual LANs and disk volumes.
3. Define version (and patch) level dependencies between constituent appliances.
4. Allow for the definition of boot order as well as dynamic startup parameters.
5. Provide management mechanisms that fit into existing management infrastructure that allows for metadata definition, deployment, oversight and diagnostics.

In addition to being comprised of virtual appliances, assemblies can also contain external references to systems that are not contained in the assembly. This is necessary to include infrastructure such as databases or JCA endpoints that cannot or should not be included in an assembly.

> “We have tested the Oracle WebLogic Server on Oracle JRockit Virtual Edition in the context of the CERN Open lab partnership and we have found impressive benefits for our organization. This solution significantly simplifies the maintenance of our middleware solutions and provides a cost-effective scalability on demand as it runs without a guest Operating System. Overall, Oracle WebLogic Server on Oracle JRockit Virtual Edition will allow us to respond faster to the request of our users”
>
> **- - Carlos Garcia Fernandez, Computer Scientist, CERN**

The Oracle Virtual Assembly Builder infrastructure and associated tooling adheres strictly to the above principles providing a comprehensive Java EE virtualization infrastructure uniquely offered by Oracle.

The Power of Assemblies

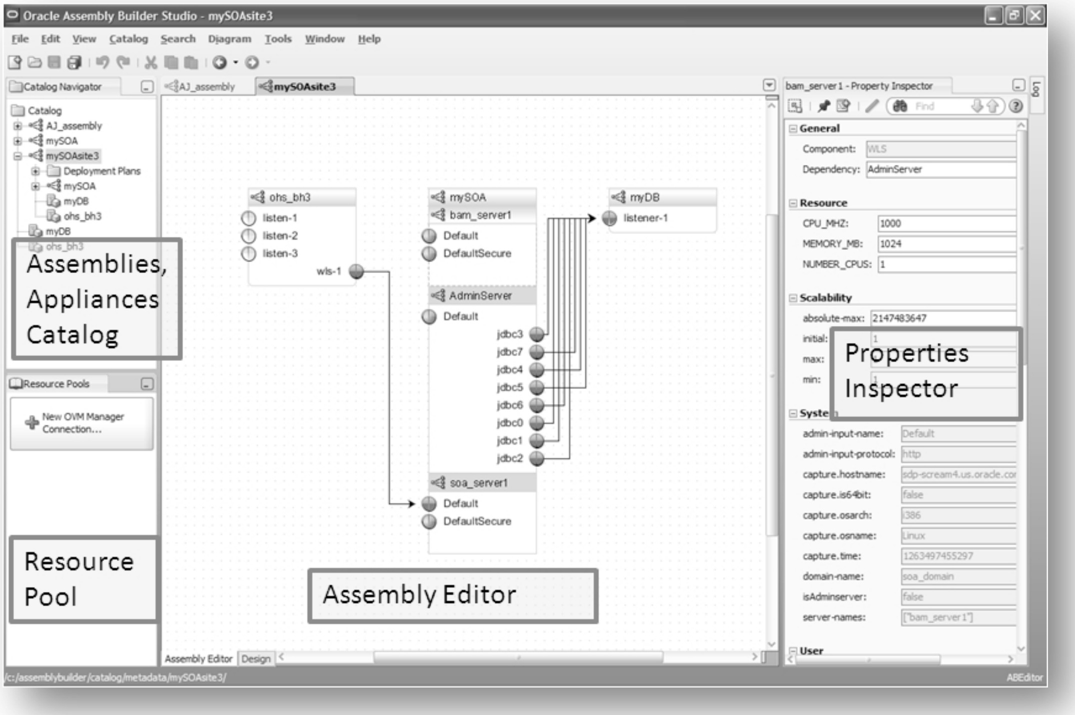
The notion of being able to create pre-built assemblies for deployment is extremely powerful and has a number of advantages that help drive down operational costs and complexity. These include:

- The ability to easily replicate assemblies in production, even allowing for variations of the assembly without adding complexity.
- Reduce risk of configuration errors as assemblies are moved between development, test and production environments.
- Replicated environments allow for high level of standardization and consistency across application infrastructures, allowing for simple implementation of best practices.
- Accelerates deployment of new infrastructures and applications.

Oracle Virtual Assembly Builder

A simple way of composing more complex assemblies of virtual appliances is required. Specifically what is needed is tooling that allows for the composition of virtual appliances as well as end point mapping of externalized systems and other larger non virtual appliance based systems such as databases and ERP systems.

Included in Oracle WebLogic Server Enterprise Edition and Oracle WebLogic Suite is Oracle Virtual Assembly Builder, a set of tooling comprised of an intuitive visual environment, as well as command line interface, allowing administrators and deployers to constitute complete assemblies encompassing all the components and systems that make up a potentially complex application structure or infrastructure. Figure 2 shows a ‘screen shot’ of Oracle Virtual Assembly Builder Studio.



General

Component: WLS
Dependency: AdminServer

Resource

CPU_MHZ: 1000
MEMORY_MB: 1024
NUMBER_CPUS: 1

Scalability

absolute-max: 2147483647
initial: 1
max:
min: 1

System

admin-input-name: Default
admin-input-protocol: http
capture.hostname: sdp-scream4.us.oracle.cor
capture.is64bit: false
capture.osarch: i386
capture.osname: Linux
capture.time: 1263497455297
domain-name: soa_domain
isAdminserver: false
server-names: ["bam_server1"]

User

FIGURE 2: ORACLE VIRTUAL ASSEMBLY BUILDER STUDIO

Oracle Virtual Assembly Builder provides the following capabilities:

- The ability to browse catalogs of existing assemblies and virtual appliances allowing for simple re-use of existing infrastructure.
- The main assembly editor that allows for declarative composition of new assemblies based on existing appliances and external systems. As depicted in Figure 7, the assembly editor section shows entry and protocol endpoints that can easily be connected between appliances using drag-and-drop.
- The properties inspector that shows the editable properties (metadata) of assembly components.
- Automated packaging of complex multi-tier applications into a collection of self-contained software appliances and assemblies.
- Analysis of application configurations and inter-dependencies.
- Creation of templatized definitions of complete configuration which allows for simple deployment.
- One-step deployment of entire multi-tier application onto virtualized resources.

The assembly creation and deployment process, depicted in Figure 3, is straight forward consisting of four simple steps. First, in the introspection phase, the necessary metadata and configuration information is captured from a physical deployment for all components that will make up the appliances within the assembly. Subsequently the relationships are established among the appliances and external resources. The packaging process simply creates the deployment artifacts necessary for the assembly that is relevant to the particular virtualization platform (e.g. virtual images). Finally, the deploy phase deploys the assembly into your development, test or production environment.

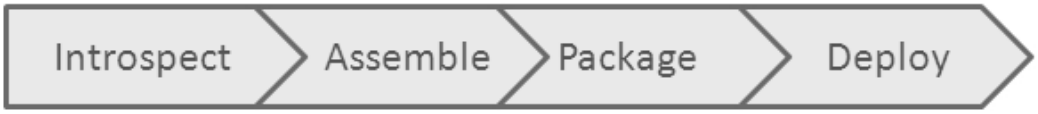


FIGURE 3: ASSEMBLY CREATION AND DEPLOYMENT PROCESS

Following the above process yields a number of benefits:

- There will no longer be a need to manually create new deployment environments. Once the virtualized pool of resources are in place, assemblies can easily be deployed without requiring additional configuration or setup.
- Over time, a catalog of virtual appliances and resources are established that will allow for efficient reuse of work.
- Oracle Virtual Assembly builder provides a convenient P2V process

The command line facility of Oracle Virtual Assembly Builder allows for complete end-to-end automation of the virtual appliance, assembly creation, configuration, provisioning and deployment process.

Oracle Virtual Assembly Builder is positioned to become Oracle's preferred mechanism for deploying applications and infrastructure onto virtualized resources.

External Resources

When creating assemblies it will often be necessary to include immovable systems within the assembly. However, it may not be desirable to package up the external system and to include in the assembly package. For this reason, Oracle Virtual Assembly Builder allows thas the ability to create and configure virtual appliances as well as introspect external systems (i.e. databases) that will not be packaged as a part of the virtual appliance. The external systems can then easily be referenced as part of the assembly just as if it were another virtual appliance creating consistency and simplicity in the configuration and deployment process.

Oracle Virtual Machine

A key building block for any virtualized deployment is a sound underlying hypervisor as well as necessary management capabilities providing provisioning, cloning, VM migration and high availability. Oracle Virtual Machine (Oracle VM) provides not only the hypervisor layer, but also the management capability as the foundation for virtualization for all of Oracle Fusion Middleware, including Oracle WebLogic Server on Oracle JRockit Virtual.

Optimizing Java EE Virtualization

Modern virtualization technology has solved a multitude of problems. However, performance continues to be an area of concern for some; particularly for Java based server side applications. There is some penalty on overall application performance of applications deployed on a hypervisor. In part this is attributed to I/O limitations in the hypervisor layer as well as CPU time slicing across potentially multiple guest operating systems and applications.

With the introduction of Oracle WebLogic Server on Oracle JRockit Virtual Edition, Oracle provides a novel way to deploy Java based middleware in a virtualized environment by allowing Java EE applications to be deployed in a virtualized environment without paying the performance and scalability penalties typically associated with running on a hypervisor.

With advances in the virtual machine layer of the Oracle JRockit virtual machine, Oracle WebLogic server can now be run directly on the hypervisor eliminating the need for a guest operating system, thereby eliminating the added overhead of running unnecessary layers of guest and host operating systems that consume valuable resources. The end result is a more secure, higher performing and better scalable virtualized deployment offering comparable performance to traditional physical deployments.

Oracle JRockit Virtual Edition

In order to understand Oracle WebLogic Server on Oracle JRockit Virtual Edition, it is necessary to understand the foundation that makes optimization of Java EE deployments possible. Oracle JRockit provides this foundation and its technical characteristics are described below.

Oracle JRockit Java Virtual Machine is the highest performing server side Java Virtual Machine and is the preferred Java Virtual Machine of customers running Oracle WebLogic Server on x86 and SPARC based hardware. JRockit comes bundled with Oracle WebLogic Server and provides excellent real time capabilities as well as tooling for diagnosing application latencies and memory leaks.

A key design goal of Oracle WebLogic Server on Oracle JRockit Virtual Edition is to provide an optimized execution environment targeted specifically at virtualized deployments. Oracle WebLogic Server interacts directly with the underlying Java Virtual Machine for all its needs (CPU, disk I/O, networking, etc). For this reason, the guest operating system required for deployment on most hypervisors becomes superfluous for all Java applications that do not make use of operating system resources directly.

With JRockit Virtual Edition, the otherwise necessary operating system services that would typically be required by the JRockit Java Virtual Machine have been implemented in JRockit Virtual Edition itself. JRockit Virtual Edition provides the necessary operating system capabilities, for the JRockit Virtual Machine to run directly on the hypervisor. One can think of this as the base services required by the most basic application (CPU, disk I/O, networking) and the Java runtime environment (JRE) providing the rest. These services have been implemented in the lightest weight possible, eliminating all overhead otherwise imposed by a guest operating system and its associated multitude of processes. Figure 4 shows a conceptual illustration of JRockit Virtual Edition hosting a Java application. Figure 4 shows the comparison of a traditional virtualized Java solution compared to the simpler solution using JRockit Virtual Edition.

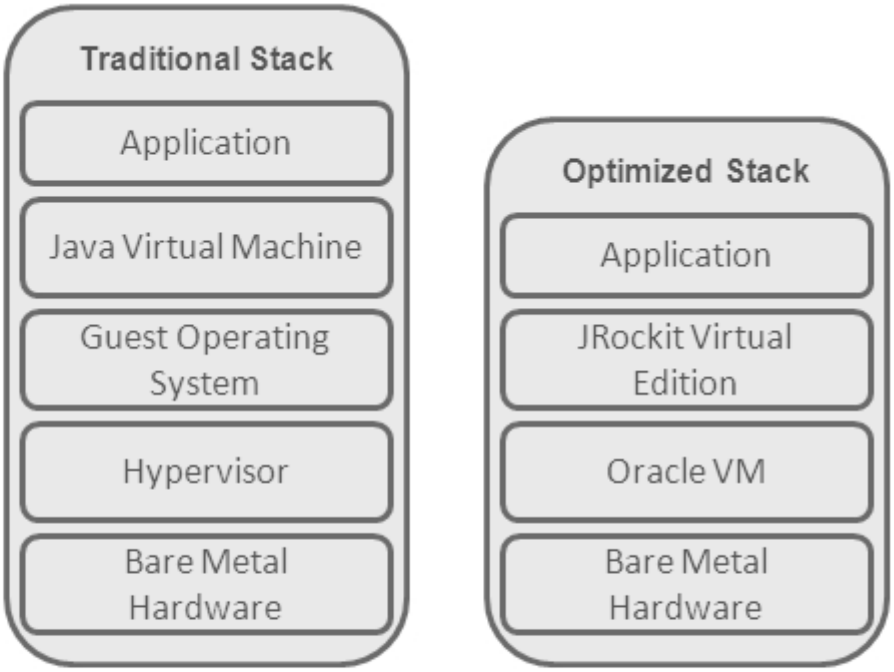


FIGURE 4: TRADITIONAL VIRTUALIZATION STACK VS. JROCKIT VIRTUAL EDITION STACK

JRockit Virtual Machine Advantages

By allowing JRockit to execute directly on the underlying hypervisor, the full potential of the virtualized hardware is completely dedicated to serve the Java Virtual Machine versus the guest operating system. The end result is better utilization of hardware, resulting in better performance and better scalability. Subsequently, this allows for more processing to take place on the exact same hardware. By freeing up resources consumed by the guest operating system it is also possible to execute additional Oracle WebLogic server instances on the same hardware compared to a traditional virtualization environment necessitating a guest operating system. Other advantages include:

- The effort of configuring[1], tuning and patching the operating system is eliminated by the inclusion of this functionality in the JRockit Virtual Edition. This results in a significant operational cost savings. Configuration of JRockit Virtual Edition is surprisingly simple, requiring only a single XML configuration file where only the basic parameters concerning CPU, Network, Virtual Disk[2] and Virtual Memory are specified. Figure 5 depicts a typical JRockit Virtual Edition configuration file. Note the simplicity with which the Java VM is configured.

[1] A typical Operating System contains thousands of configuration files.

[2] The Virtual Disk specific is a NFS mount point.

```
<?xml version="1.0" encoding="UTF-8"?>

<jrockitve-imagetool-config

xsi:noNamespaceSchemaLocation="jrockitve-imagetool-config.xsd"

  <jrockitve-config memory="256 MB" cpus="1">

      <disks>
        <disk id="root" size="180 MB"/>
      </disks>
      <mounts>
        <mount>
          <mount-point>/</mount-point>

        </mount>
      </mounts>
    </storage>
    <vm-name>default-vm</vm-name>
    <java-arguments>HelloWorld</java-arguments>
    <network>
      <nics>
        <nic type="bridged"/>
      </nics>
    </network>


url>
</jrockitve-imagetool-config>
```

FIGURE 5: SAMPLE JROCKIT VIRTUAL EDITION CONFIGURATION FILE

- The cost and effort of making an operating system secure should not be overlooked. A significant amount of an administrators time is spent consuming and evaluating various security bulletins to understand the impact on their respective operational environments. By eliminating the guest operating system, the largest source of security concern has been eliminated. Thereby, allowing operational staff to eliminate this work item from their daily activities.
- Administrators can focus more on application rather than operating system administration.
- Application patching becomes even simpler with JRockit Virtual Edition by providing support for offline configuration and patching, something that is not possible on traditional operating systems including Linux, UNIX and Windows.
- Booting a JRockit Virtual Machine instance occurs in seconds compared to minutes with a traditional virtualization system involving a guest operating system, significantly accelerating provisioning. This allows for a more dynamic deployment environment capable of rapidly adapting to user demands.

Oracle WebLogic Server on Oracle JRockit Virtual Edition

Oracle WebLogic Server on Oracle JRockit Virtual Edition provides a complete and optimized virtualized runtime environment for Java EE applications. Figure 6 provides a complete illustration of the Oracle WebLogic Server on Oracle JRockit Virtual Edition execution stack.

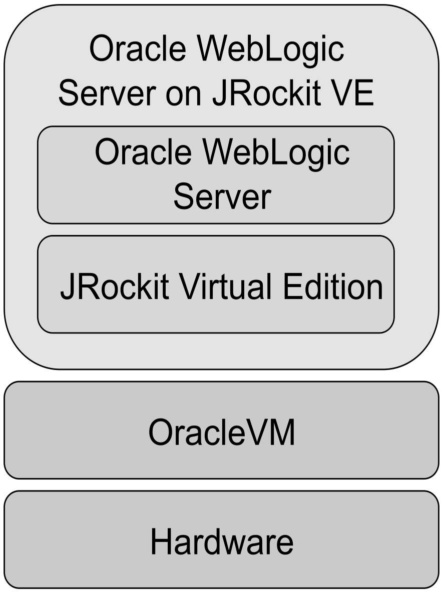


FIGURE 6: ORACLE WEBLOGIC SERVER ON ORACLE JROCKIT VIRTUAL EDITION EXECUTION STACK

Although a new deployment paradigm, users familiar with WebLogic Server application development, deployment and administration will be able to make use of existing knowledge, tools and techniques. The application development and deployment process remains consistent with the traditional non-virtualized deployment paradigm. Virtual appliances can easily be created based on a WebLogic Server domain. These domains can be pre-populated with applications already deployed, or can be generic with applications deployed to WebLogic Server once it is up and running in the virtualized environment.

As for administration, WebLogic Server administration tools such as WebLogic Scripting Tool (WLST) scripts, the WebLogic Server Management Console and weblogic.Deployer can still be used to administer and manage Oracle WebLogic Server on Oracle JRockit Virtual Edition. There have been no runtime changes to the WebLogic Server code base ensuring a low risk transition to Oracle WebLogic Server on Oracle JRockit Virtual Edition. In addition, monitoring and diagnostic tools such as AD4J and JRockit Mission Control continue to function as they did before in a non-virtualized deployment environment.

Oracle WebLogic Server on Oracle JRockit Virtual Edition provides a simple command line tool (wlsveimagetool.jar) that allows for the creation of WebLogic Server based virtual appliances. The functionality by the Image Tool can be scripted using shell scripts and [Apache] Ant allowing for end-to-end automation of the virtual appliance creation and provisioning process.

Once a WebLogic Server based virtual appliance has been created, the appliance can simply be deployed on the physical servers using Oracle VM. Maintenance of the virtual appliance can be performed using standard WebLogic management tools for configuration, deployment and lifecycle management.

Performance

Although virtualization has been a key technology for reducing hardware cost, virtualization has imposed such overhead making it unviable for production deployment. With Oracle WebLogic Server on Oracle JRockit Virtual Edition, Oracle has overcome this performance hurdle for Java EE based applications.

Performance benchmarks of Oracle WebLogic Server on Oracle JRockit Virtual Edition shows a significant performance improvement over traditional virtualized deployments. Table 1 describes the configuration of a performance test scenario using identical hardware. The data illustrates a significant 30% improvement in overall throughput of the application compared to the traditional para-virtualized deployment, while achieving 83% performance of a physical deployment. This translates into higher application density which is one of the goals of virtualization that contributes to an overall cost reduction. It is also important to note that the Dom0 (domain 0) CPU utilization shows a 26% reduction versus the traditional virtualized environment. The latter indicates that it is possible to execute additional instances of Oracle WebLogic Server on Oracle JRockit Virtual Edition without performance penalty as a result of exceeding the Dom0 CPU.

CONFIGURATION	PHYSICAL O/S	ORACLE VM / WLSVE	GUEST O/S PARA VIRTUALIZED
WEBLOGIC SERVER	10.3.1 GA	10.3.1 GA	10.3.1 GA
CLOCK SPEED	2.93 GHz	2.93 GHz	2.93 GHz
MEMORY	24GB	4GB	4GB
MEMORY SPEED (MHZ)	1066	1066	1066
JROCKIT	R27.6.2-20	R27.6.3-40	R27.6.2-20
HEAP	3600M	3400M	2600M
OPERATIONS / SEC	335.94	306.59	249.09
DOM0 CPU UTIL.	--	28.40%	39.51%

TABLE 1: SAMPLE PERFORMANCE COMPARISON DATA

Additional performance metrics are in the process of being published.

When Not to Use Oracle WebLogic Server on Oracle JRockit Virtual Edition

As most Java EE applications do not make explicit OS calls, Oracle WebLogic Server on Oracle JRockit Virtual Edition becomes an attractive deployment alternative for virtualized environments. However, there are some restrictions on the applications deployed on top of Oracle WebLogic Server on Oracle JRockit Virtual Edition that need to be understood. These include:

- Deployed Java EE applications cannot rely upon 'System' operating system calls as these will fail.
- Applications can not contain native code, since JRockit Virtual Edition is optimized for running pure Java applications.

Summary

Oracle WebLogic Server on Oracle JRockit Virtual Edition running directly on Oracle VM raises the bar for virtualized Java EE deployments by creating an optimized server side Java Virtualization stack. By eliminating the guest operating system, there are several key advantages that are realized:

- Better Hardware Utilization resulting in better overall performance and better scalability of the hosted Java EE application.
- Integration with Oracle Virtual Assembly Builder delivers simplified deployment of entire WebLogic Server domain onto virtualized resources
- Simple provisioning through Oracle VM infrastructure[3].
- Offline maintenance and patching not available with a traditional guest operating system.
- Reduced security vulnerability

Oracle WebLogic Server on Oracle JRockit Virtual Edition creates not only a novel runtime paradigm for Java EE deployments, but also combines this environment with Oracle Virtual Assembly Builder and runtime management and operational tooling into a complete virtualization solution. The end result is virtualization infrastructure optimally suited to reduce deployment and management complexity resulting in increased hardware efficiency and utilization while at the same time drastically reducing operational costs.

[3] Enterprise Manager Grid Control support will become available in a future release of WebLogic Suite.

White Paper Title
April 2010 Rev A.
Author: Erik Bergenholtz
Contributing Authors: Mark Prichard

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com