

Consolidation Using Oracle's SPARC Virtualization Technologies

ORACLE TECHNICAL WHITE PAPER | OCTOBER 2015





Table of Contents

Introduction	3
Designing a Consolidated Infrastructure	6
Seven Areas of Consideration for Consolidation	6
Security Isolation	6
Resource Isolation	6
Workload Efficiency	7
Availability	7
Serviceability	7
Flexibility	8
Agility	8
Requirements-Based Consolidation	9
Oracle Virtualization Technologies	9
Physical Domains (PDoms)	9
Oracle VM Server for SPARC	10
Control, I/O, Service, Guest, and Guest Root Domain Roles	11
Guest Domains Model	11
Redundant Guest Domains Model	12
SR-IOV or Direct I/O Domains Model	13
Redundant SR-IOV Domains Model	14
Guest Root Domains Model	15
Oracle Solaris Zones	16
Native Branded Zones	17



Kernel Zones	18
Non-Native Branded Zones	19
Combining Virtualization Technologies	20
Redundant Guest Domains and Oracle Solaris Zones	22
Guest Root Domains and Oracle Solaris Zones	23
Root Domains and SR-IOV Domains	24
Hybrid Combination of All Oracle Virtualization Technologies	26
Summary of Characteristics for Combined Virtualization Technologies	27
Conclusion	28
About Oracle Elite Engineering Exchange	28



Introduction

This paper provides a high-level overview of Oracle's virtualization technologies, and it introduces a methodology for evaluating their features so that they can be matched against workload requirements by observing the following seven characteristics:


- » Security isolation
- » Resource isolation
- » Efficiency
- » Availability
- » Serviceability
- » Flexibility
- » Agility

This methodology could also be used to evaluate other Oracle virtualization technologies, as well as other combinations of Oracle virtualization technologies not covered in this paper, such as pluggable databases in Oracle Database 12c or application consolidation within Oracle WebLogic Server.

All organizations have a requirement to run their IT infrastructure in the most efficient way possible. The current trend is to use virtualization technologies to allow the consolidation of many diverse workloads onto a smaller number of physical servers, thereby achieving cost savings. This trend is also driven by the increased capacity of modern servers compared to some of the workloads, and the requirement to consolidate those workloads to avoid the massive underutilization of those servers that would result if a single-workload-per-server route was followed.

One of the most important aspects of any consolidation exercise is to understand exactly where the costs are coming from, and to build an architecture that is focused on minimizing the total cost of ownership (TCO) of the data center infrastructure. The benefits of consolidation manifest themselves in the following areas:

- » Operations: The infrastructure is simplified due to a reduction of the number of diverse objects that need to be monitored, managed, and maintained.
- » Deployment: Standards-based pools of resources allow easier deployment of new workloads by avoiding the need to procure and implement new infrastructure for each new project.
- » Infrastructure: Consolidation results in higher utilization of hardware and software assets and allows a smaller infrastructure footprint, which reduces the capital costs of acquiring those assets.




The primary goal of consolidation is cost reduction, and the cost of operating and managing the architecture is one of the largest components of cost within a data center. For this reason, any consolidation exercise should focus on driving operational savings. This is generally achieved by simplifying the architecture, by reducing the number of entities that need to be managed, and by standardizing the operating system (OS), patch levels, and computational building blocks so the components can all be managed in exactly the same way.

Some of the models described in this paper enforce this standardization, because they require workloads to share the same OS version. The benefits are that standardization reduces the number of OS versions that need to be independently managed, it reduces the overall quantity of OS instances, and—by extension—it reduces the size of the underlying infrastructure to support them. Oracle Solaris supports this type of consolidation very well due to the Application Binary Compatibility guarantee, which means that, in essence, applications are not tied to a specific patch level or kernel release. This removes the need to maintain a multitude of unique Oracle Solaris versions per application stack.

When a traditional monolithic virtualization approach is taken where machines are mapped one-to-one to virtual machines, there is no overall reduction in the operational complexity of the system, because there are still the same number of entities to be managed, plus there is additional overhead to manage the newly introduced virtualization infrastructure as well. The aim should be to consolidate workloads, not simply to consolidate machines, because it is workload consolidation that will drive the operational efficiencies of the data center.

The choice of the most appropriate virtualization technology needs to be driven by requirements. The requirements of a development environment are likely to be substantially different from those of a mission-critical production environment. Oracle Solaris 11.2 and above provides the capability—through the use of its Unified Archive feature—to seamlessly migrate a workload among any of the deployment options. It is, therefore, possible to adopt different virtualization models for development, test, and production environments, and be able to migrate the same workload onto any deployment type. More importantly, as the production environment evolves over time, Unified Archives provide the flexibility to move the workload to the most appropriate deployment type as well.

This paper will show that a combination of Oracle virtualization technologies allows the most appropriate deployment choice to be made, based on the workload requirements. The flexibility of the models and workload deployment options allows the architecture to be easily modified to meet changing requirements over time as well.



This white paper assumes that the reader has a working understanding of Oracle's virtualization technologies. There are numerous white papers that discuss the technical aspects of each of these technologies, including specific implementation guidelines.

Specifically, the white paper "Oracle's SPARC M7 and SPARC T7 Servers: Domaining Best Practices" covers physical domains (PDoms) and Oracle VM Server for SPARC (formerly called Sun Logical Domains) technology in detail, and it should be considered prerequisite reading for this white paper.



Designing a Consolidated Infrastructure

When faced with multiple options for consolidation, it is useful to remember the following reasons for consolidating in the first place, and use those initial requirements to derive the most appropriate solution:

- » Maximize operational efficiency
 - » The benefits of consolidation are not purely derived from a reduction in hardware cost. The majority of the consolidation benefits are derived from the simplicity that accrues from standardization of an operating model and the reduction in the number of managed objects.
 - » Consolidating as high up the stack as possible naturally reduces the total number of managed objects, as well as creates as much standardization as possible.
- » Maximize workload efficiency
 - » One of the trade-offs of increased isolation is a potential increase in the virtualization overhead. Bear this in mind, and create additional isolation only where necessary.
 - » Some workloads are quite small in comparison to the footprint of a modern OS instance. Try to co-locate multiple workloads per OS instance where possible.

Each of the Oracle virtualization technologies has different characteristics, and understanding how they are different enables you to match the application and workload requirements with the correct combination of Oracle technologies.

Seven Areas of Consideration for Consolidation

A consolidated data center has significantly different characteristics than a siloed data center, and these differences require a change in the way that security, resource allocation, availability, and serviceability are all managed.

Security Isolation

In a siloed environment, discrete hardware allows workloads to be isolated purely by virtue of the physical separation. Consolidated environments will have a large number of shared components, and new technologies might need to be introduced to guarantee the level of separation that is required. In most cases, existing security policies will need to be revisited to accommodate the new virtualized, shared environments. In other cases, it might be necessary to adopt particular virtualization technologies to abide by existing policies.

Resource Isolation

Workloads with dedicated hardware have guaranteed resource allocation, but this results in large amounts of underutilized resources. One of the main benefits of consolidation is to pool all the compute resources so that they can be used more efficiently. When resources are pooled in this way, measures need to be put in place to ensure that each of the workloads has access to sufficient resources to be able to deliver the required service levels. At the other end of the scale, it is imperative that a single workload should not be able to affect the other workloads in the pool through excessive consumption of resources.

Workload Efficiency

Most virtualization technologies create some level of overhead. This usually manifests itself in three distinct ways:

- » Hypervisor overhead: This is the amount of CPU and memory resources that has to be allocated to run the hypervisor. The resources consumed depend on the number of virtual machines (VMs) being managed and their I/O characteristics. Typically, as the number of workloads grows, the amount of resources that must be allocated to the hypervisor also grows.
- » Virtualized resource overhead: The hypervisor is usually responsible for allocating virtual CPUs and memory to the VMs. It also provides virtual I/O resources to the VMs. The actual performance of the virtual CPUs compared to the physical CPU is dependent on the level of oversubscription and is usually lower; the same is often true for memory. In most cases, the throughput and latency of virtual I/O are not as good as the throughput and latency of native I/O. This reduction in performance needs to be taken into consideration when sizing the physical infrastructure for consolidated workloads.
- » OS overhead: It is also worth considering that modern operating systems are themselves quite heavy consumers of resources. That is, each individual OS instance needs its own boot device, disk, and I/O allocation, as well as a CPU and memory pool, some of which it consumes for its own housekeeping tasks. When a single workload is deployed per OS instance, quite often the OS instance itself consumes more resources than the workload consumes. This is often referred to as the *workload-to-payload ratio*. Better efficiency can be obtained by consolidating as many workloads per OS instance as possible, either by using application-level consolidation or by using OS virtualization tools.

The combination of the first two effects is colloquially known as *virtualization tax*. The third effect could be considered a tax rebate when the number of OS instances and the associated resource and management overhead are reduced as part of the consolidation exercise.

Availability

Dealing with a large number of single-workload servers has the advantage of reducing the impact of a server failure because a failure usually affects only a single service or workload. As more workloads are consolidated onto a smaller number of servers, the impact of a server outage (whether planned or unplanned) is greater. This means that the overall architecture has to consciously focus on ensuring that the resulting increased availability requirements are met. Usually this involves implementing high availability (HA) solutions where they might not previously have been considered necessary for single workloads, as well as ensuring that the underlying hardware has higher reliability, availability, and serviceability (RAS) features, so that failures at the hardware layer do not immediately trigger an outage.


While there is an initial design overhead to putting an HA environment in place, it is often possible to heavily leverage such an environment using multiple consolidated virtual environments at no or minimal additional cost, with significant uptime improvements across multiple virtualized environments.

When consolidating workloads, it is always helpful to consider the entities within the architecture that could fail, the likelihood of that failure, and the impact that failure could have on the running workloads.

In the comparisons presented later in this document, the availability score is made under the assumption that additional HA derived from clustering technology is not applied. In all cases, the availability characteristic will increase by applying clustering technology.

Serviceability

Similar to the heightened availability requirements, an outage that is required to service a component impacts a much greater proportion of workloads. It is difficult enough to negotiate an outage window for a single production workload, but trying to simultaneously arrange an outage window for increasing numbers of workloads becomes exponentially difficult.



This serviceability problem can be solved by minimizing the number of service events that impact the running workloads. In many cases, the availability architecture outlined above can fulfill many of the serviceability requirements.

Flexibility

In any environment, the ability of the system administrator to plan for the changing real-time demands of the workload can be difficult, but in a consolidated, virtualized environment this problem is exacerbated. Now, rather than sizing and planning for a single workload, the administrator must look at the requirements of all of the consolidated workloads.

This is further complicated by the dynamic nature of most workloads, which might vary from being quiescent and asserting almost no load, to fully consuming all of the allocated resources. When multiple workloads are consolidated on a single platform, rightsizing can become a difficult problem to solve. The simple approach of over-allocating resources to cope with peaks in workload is inefficient and results in an artificial increase in the virtualization tax discussed earlier.

The tools and technologies used to manage virtualization must provide as much flexibility as possible to dynamically change the resource allocations of a virtualized environment to avoid situations where a quiescent load is allocated (but not using) significant platform resources, while another heavily loaded environment is unable to perform due to insufficient resources in its environment. Dynamically moving resources from quiet environments to busy ones can enable better overall utilization of resources and system response to workloads with wide variances in utilization.

The tools provided to allow this flexibility vary according to the virtualization technology used, but at all levels some granularity of dynamic resource reallocation is possible, and in some cases it can be preconfigured according to policies that allow environments to automatically grow and shrink according to predefined rules that have been established by the system owner.

The flexibility to scale workloads up and down (or even off) is invaluable in maximizing utilization and throughput. It should be noted that this flexibility to resize environments on the fly has potential consequences, such as possible impacts on licensing requirements for the software deployed into the environment.

Making careful choices concerning which virtualized workloads share a virtualized platform and the policies used to govern their allocation of resources can result in excellent overall platform utilization levels and the minimization of unused resources and wasted capital investment.

Agility

When many workloads are running on a server within a pool, and that server needs to be taken out of service for maintenance, it is important to be able to move those workloads with minimal or zero service disruption to an alternative server for the duration of that maintenance. As the number of workloads increases, the difficulty of negotiating a simultaneous outage window also increases.

The agility characteristic measures the ease of workload migration from one location to another, in terms of the impact, complexity, and duration of the migration.

It should be noted that many modern workloads are designed to run in a scalable fashion, which means that individual workload instances can easily be turned off without any impact to service, for instance, with Oracle Real Application Clusters (Oracle RAC) or Oracle WebLogic Suite. The agility characteristic is important for workloads that do not have this capability.



Requirements-Based Consolidation

Every virtualization technology has different characteristics for isolation, overhead, efficiency, and flexibility. The various workloads required to run on these technologies have different requirements as well. It should be clear that one size almost never fits all, and that the most effective virtualization choice is to use the right tool (or combination of tools) for the right workload. The following sections discuss each of Oracle's virtualization technologies in terms of the seven characteristics outlined above.

Oracle Virtualization Technologies

Oracle has a number of virtualization technologies that operate at different layers of the stack, which provides the ability to build a consolidated infrastructure to maximize the cost savings outlined previously:

- » Physical domains (PDOMs): Oracle's SPARC M7-8, M7-16 and SPARC M6-32 servers provide the option to divide a large server into a number of smaller ones by physically segregating the CPU, memory, and I/O resources into a number of PDOMs. Each PDOM behaves just like a physical server.
- » Oracle VM Server for SPARC logical domains (LDOMs): All current SPARC servers from Oracle (or PDOMs, as described above) can be divided into a number of smaller systems by logically segregating CPU and memory resources and providing either native or virtualized I/O to domains. This adds an additional layer of granularity, but also has the advantage of allowing dynamic reallocation of CPU and memory resources among the LDOMs. Oracle VM Server for SPARC technology provides a large amount of choice for the configuration of the LDOMs, and this paper discusses a number of common scenarios. Every PDOM or SPARC server currently offered by Oracle always has at least one LDOM that initially owns all the resources. Oracle VM Server for SPARC is part of the Oracle VM product family, which includes Oracle VM Server for x86. Oracle VM Server for SPARC and Oracle VM Server for x86 share several architectural principles, especially the use of privileged domains for system control, and can be administered with Oracle VM Manager, but that is not in the scope of this white paper.
- » Oracle Solaris runs in every LDOM and includes a feature called Oracle Solaris Zones that provides the ability to create a number of additional virtualized OS instances. This provides the finest-grained allocation of resources in an extremely efficient manner. Similar to LDOMs, Oracle Solaris Zones technology is extremely flexible and allows a large number of different deployment choices, which are discussed in this paper.

Physical Domains (PDOMs)

PDOMs enable electrically isolated server hardware, which means administrators can isolate hardware or security faults and constrain their exposure to each domain. The result is a superior level of system availability and security. This technology is available in the SPARC M7 and SPARC M6-32 servers.

Software and hardware errors and failures do not propagate beyond the domain in which the fault occurred. Complete fault isolation between PDOMs limits the effect on applications of any hardware or software errors. This helps to maintain a high level of availability in these servers, which is necessary when consolidating many applications. Each PDOM is administered separately, so a security breach in one domain does not affect any other domain.

PDOM technology allows a large server to be split up into a number of smaller fully isolated servers. The main reason for doing this is usually to create smaller building blocks for workload isolation or to improve serviceability. In effect, each PDOM can be considered to be a standalone physical server. For Oracle software licensing purposes, a PDOM is considered a hard partition.

Figure 1 and the following list describe how PDOMs address the seven characteristics outlined earlier:

- » **Security isolation:** Each PDom is, in effect, a fully isolated physical server with fully dedicated components. The security characteristics of PDom are identical to those of a standalone server, although PDom do share a common administrative access via the Oracle Integrated Lights Out Manager (Oracle ILOM).
- » **Resource isolation:** Each PDom has fully dedicated CPU, memory, and I/O resources. Its resources are fully isolated from the resources of all other PDom within the server.
- » **Efficiency:** All workloads run at bare-metal performance levels with neither hypervisor nor virtualized resource overhead, as defined in the “Workload Efficiency” section above.
- » **Availability:** A PDom is usually constructed with fully redundant components, including I/O resources, which means that it can usually accommodate single failures without interruption.
- » **Serviceability:** The SPARC M7 and SPARC M6-32 servers allow the hot plugging of I/O cards within the system, so that individual cards can be serviced. However, servicing of CPU and memory components requires an outage.
- » **Flexibility:** A PDom should be considered a fixed hardware resource. CPU, memory, and I/O resources cannot be dynamically reallocated between PDom within a system.
- » **Agility:** Due to the physical nature of all the resources in a PDom, it is not possible to migrate PDom between systems.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
Very High	Very High	Extremely High	Extremely High	Extremely High	Low	Low

Figure 1. Summary of the seven characteristics for PDom

Oracle VM Server for SPARC


An Oracle VM Server for SPARC domain (also referred to as a logical domain or LDom) is a virtual machine that comprises a discrete logical grouping of resources. An LDom has its own operating system and identity within a single computer system. A variety of application software can be run in different LDom and kept independent for performance and security purposes. This technology is available on every SPARC server currently offered by Oracle.

For Oracle software licensing, an LDom is considered a hard partition if it is configured appropriately.

This technology most closely resembles the traditional virtualization technology used with the concept of a hypervisor and guest VMs, but it is implemented in a very different way. Each LDom is only permitted to observe and interact with those server resources that are made available to it by the hypervisor. The hypervisor enforces the partitioning of the server’s resources and provides limited subsets to multiple operating system environments. This partitioning and provisioning is the fundamental mechanism for creating LDom.

Each LDom can be managed as an entirely independent machine with its own resources, such as:

- » Dedicated CPU and memory resources
- » Kernel, patches, and tuning parameters
- » User accounts and administrators
- » Disks
- » Network interfaces, media access control (MAC) addresses, and IP addresses



Each LDom can be stopped, started, and rebooted independently of other LDoms without requiring users to perform a power cycle of the server and without affecting the other running LDoms.

Control, I/O, Service, Guest, and Guest Root Domain Roles

A number of different names are used for the various roles of domains that can exist in an Oracle VM Server for SPARC deployment. This is complicated by the fact that a domain can be of more than one type simultaneously. For example, a control domain is always an I/O domain, and it is usually a service domain. For the purposes of this paper, the following terminology is used to describe the different Oracle VM Server for SPARC domain types:

- » **Control domain**—The management control point for virtualization of the server, which is used to configure domains and manage resources. It is the first domain to boot on a power-up, is an I/O domain, and is usually a service domain as well. There can be only one control domain.
- » **I/O domain**—A domain that has been assigned physical I/O devices. It can be a PCIe root complex and associated devices or PCIe slots, a PCIe device, or a single-root I/O virtualization (SR-IOV) function. It has native performance and functionality for the devices it owns, unmediated by any virtualization layer. There can be multiple I/O domains.
- » **Service domain**—A domain that provides virtual networking and disk devices to guest domains. There can be multiple service domains. A service domain is always an I/O domain, because it must own physical I/O resources in order to virtualize them for guest domains. In most cases, these service domains have PCIe root complexes assigned to them, and they could be called a root domain in this case.
- » **Guest domain**—A domain whose devices are all virtual rather than physical. Virtual networking and disk devices are provided to a guest domain by one or more service domains. In common practice, a guest domain is where applications are run. There usually are multiple guest domains in a single system.
- » **Guest root domain**—A domain that has one or more PCIe root complexes assigned to it, but is used to run applications within the domain, rather than to provide services such as the service domain does. Physically there is no difference between service domains and guest root domains other than their usage, and guest root domains often will be simply referred to as root domains.

There are, broadly speaking, four deployment models that are typically used when running Oracle VM Server for SPARC, although hybrid models are also possible with a mix of all types of guest domains:

- » Guest Domains model
- » Redundant Guest Domains model
- » SR-IOV or Direct I/O Domains model
- » Guest Root Domains model

These different deployment models are described in detail in numerous white papers and webcasts available at this [Oracle VM Server for SPARC web page](#), but the following sections summarize the four main deployment models.

Guest Domains Model

In this model, the control domain owns *all* the root complexes and creates virtual devices for all the guest domains. This is the most flexible model, and it fits cases where there are large numbers of relatively small domains, with low impact from failure. All guest domains are affected by an outage of the control domain. Live migration is possible between guest domains. Guest domains are useful also for lightweight production environments where availability is provided by horizontal scaling at the application tier, and they are a particularly good fit for test and development environments, where performance and availability are not critical requirements.

Figure 2 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** Each guest domain acts like a physical server with its own version of the operating system. However, all I/O passes through the control domain and is usually configured to use shared physical devices.



Security can be configured within the control domain and in the guest domain itself to protect against eavesdropping and unauthorized access.

- » **Resource isolation:** Each guest domain has dedicated CPU and memory resources, which guarantees access to those resources. The virtualized I/O is provided by the control domain, usually over shared I/O devices, and it is possible to have resource contention at this level. Configuring additional virtual device services and using resource controls within the control domain can usually mitigate this contention. If necessary, different guest domains can be allocated to different physical I/O devices as well.
- » **Efficiency:** Oracle VM Server for SPARC technology allows the guest domains to make use of CPU and memory resources directly without any virtualization overhead. However, this model requires the allocation of resources to the control domain to provide the virtualized I/O functionality, and there is also a small performance overhead for virtualized I/O. This model has hypervisor and virtualized resource overhead, but that applies only to the I/O components of the workload.
- » **Availability:** The guest domains are completely reliant on the control domain, so if the control domain fails, all the guest domain's I/O will freeze until the control domain has been restored. Depending on the requirements, the guest domains can be clustered with domains on other physical servers. The I/O can be configured within the control domain to provide redundant paths to the network and disks, so that failures of this type will have no impact to the guest domain.
- » **Serviceability:** If the control domain requires servicing, the guest domains will need to also have an outage, or they will need to be migrated to other servers. However, the guest domains can be serviced without affecting any other guest domains.
- » **Flexibility:** Guest domains are highly flexible. They can be dynamically created, resized, or destroyed relatively easily. Policies can be configured in the control domain to automatically expand and contract the size of a guest domain based on its workload.
- » **Agility:** Guest domains can be live-migrated or cold-migrated to other SPARC servers. This provides a capability to migrate workloads with little or no service disruption.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	Very High	Medium	Medium	Medium	High	High

Figure 2. Summary of the seven characteristics for the Guest Domain model

Redundant Guest Domains Model

In this model, redundant I/O services are provided by a pair of service domains (one of which is the control domain). The characteristics are very similar to those for the guest domain model, except that the guest domains are not affected by a control domain failure or a service domain failure. This model is good for production environments where higher availability is required. There is additional overhead for the resources that must be dedicated to the additional service domain.

Figure 3 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** Each guest domain acts like a physical server with its own version of the operating system. However, all I/O passes through the control and service domains, and is usually configured to use shared physical devices.

Security can be configured within the control and service domains and in the guest domain itself to protect against eavesdropping and unauthorized access.

- » **Resource isolation:** Each guest domain has dedicated CPU and memory resources, which guarantees access to those resources. The control and service domains provide the virtualized I/O, usually over shared I/O devices, and it is possible to have resource contention at this level. Configuring additional virtual device services and using resource controls within the control and service domains can usually mitigate this contention. If necessary, guest domains can be allocated to different physical I/O devices.
- » **Efficiency:** Oracle VM Server for SPARC technology allows the guest domains to make use of CPU and memory resources directly without any virtualization overhead. This model has higher hypervisor overhead than the previous model, because it uses two domains to provide redundant I/O services, each of which needs to have resources allocated to it. There is also a small performance overhead for virtualized I/O. This model has hypervisor and virtualized resource overhead, but that only applies to the I/O components of the workload.
- » **Availability:** In contrast to the Guest Domain model, these domains are no longer fully reliant on the control domain, so both the control and service domains would need to fail for guest domains to be interrupted. There are still failure scenarios that could cause the whole server to fail, causing all domains to fail. The I/O should be configured between the control and service domains to provide redundant paths to networking and disk resources, so failures will have no impact to the guest domain.
- » **Serviceability:** High serviceability is one of the main benefits of the multiple-service-domain approach in that the control and service domains can be serviced without affecting the running guests, and the guests themselves can be independently serviced.
- » **Flexibility:** Guest domains are highly flexible. They can be dynamically created, resized, or destroyed relatively easily. Policies can be configured in the control domain to automatically expand and contract the size of a guest domain based on its workload.
- » **Agility:** Guest domains can be live-migrated or cold-migrated to other SPARC servers. This provides a capability to migrate workloads with little or no service disruption.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	Very High	Medium	Very High	High	High	High

Figure 3. Summary of the seven characteristics for the Redundant Guest Domains model

SR-IOV or Direct I/O Domains Model

Oracle VM Server for SPARC allows the domain controlling a physical device to provide direct access to that device to a guest domain. The two methods for accomplishing this are to use either direct I/O—where a local motherboard or PCIe device/card/slot is assigned to be a guest—or to use SR-IOV—where the card itself (the “physical function,” or PF) supports virtualization, and a virtual function (VF) can be directly assigned to the guest. These techniques allow the guest domain direct access to the physical device. It should be noted that the guest domain is still dependent on the I/O domain that owns the physical device. Multiple I/O domains can be configured to provide SR-IOV devices to different groups of guest SR-IOV domains. It should be noted that the Oracle servers based on the SPARC M7 processor do not support direct I/O. The abundance of root complexes, and the availability of PCIe cards that support SR-IOV technology arguably make the use of direct I/O technology obsolete.

The main benefit of this approach is that guests operate at near bare-metal performance levels for I/O. Additionally, this permits smaller service domains since fewer resources are needed to drive virtual I/O, thus freeing up resources for guests. It also permits access to device types not available through the virtual network and disk services. This method supports a large number of LDomS, but this is limited by the number of VFs that the I/O cards will support. In addition, live migration is not possible for guests using SR-IOV or direct I/O - they must be removed from the domain before migration.

Figure 4 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** Each guest domain acts like a physical server with its own version of the operating system. In most cases, only some of the I/O is configured “directly”; there is still some virtual I/O, and there is still usually some I/O traffic via the control or service domains.

Security can be configured within the control and service domains and in the guest domain itself to protect against eavesdropping and unauthorized access.

- » **Resource isolation:** Each guest domain has dedicated CPU and memory resources, which guarantees access to those resources. The SR-IOV and direct I/O allow each guest domain to have direct access to the physical device, although in the case of SR-IOV, the physical device is itself virtualized, and it is possible to have some resource contention, but no more than if the workloads were running natively on the same shared device.
- » **Efficiency:** Oracle VM Server for SPARC technology allows the guest domains to make use of CPU, memory, and I/O resources directly without any virtualization overhead. This model also requires the allocation of resources to the service domain, but it requires fewer resources than the previous two models where the service domain was required to provide virtualized I/O services to the guest domains. In summary, this model almost completely removes the virtualization resource overhead, but some hypervisor overhead is still present.
- » **Availability:** The use of either direct I/O or SR-IOV places a dependence on the domain that owns the physical device. If the controlling domain has an unplanned failure, the failure will also affect the guests owning the dependent devices and result in undefined consequences—usually a panic. In some cases, this might result in a longer recovery time than with other failure modes.
- » **Serviceability:** If the domain that owns the physical device requires servicing, the guest domains will also need to have an outage or be migrated to other servers. However, the guest domains can be serviced without affecting any other guest domains.
- » **Flexibility:** SR-IOV domains are highly flexible. They can be dynamically resized, created, or destroyed relatively easily. They are slightly less flexible than guest domains due to the direct mapping of physical I/O to the domains.

Agility: Domains using non-virtualized I/O cannot be live-migrated, but they can be cold-migrated to other SPARC servers. This provides a capability to migrate workloads with relatively low service disruption. The target server must provide the same underlying I/O infrastructure, which makes this process slightly more complicated than with the Guest and Redundant Guest Domain models.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	Very High	High	Medium	Low	Medium	Medium

Figure 4. Summary of the seven characteristics for the SR-IOV or Direct I/O Domains model

Redundant SR-IOV Domains Model

Similar to the Redundant Guest Domains model, and starting with Oracle VM Server for SPARC 3.2, it is possible to configure redundant I/O domains providing SR-IOV physical functions (PFs) that can be used to provide virtual functions (VFs) to guests.

The main benefit of this approach is that guests operate at near bare-metal performance levels for I/O, as well as gaining the availability and serviceability provided by resilient I/O domains. This method supports a large number of LDoms, but is limited by the number of VFs that the I/O cards will support.

Figure 5 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** Each guest domain acts like a physical server with its own version of the operating system. In most cases, only some of the I/O is configured “directly”; there is still some virtual I/O, and there is still usually some I/O traffic via the control or service domains.

Security can be configured within the control and service domains and in the guest domain itself to protect against eavesdropping and unauthorized access.

- » **Resource isolation:** Each guest domain has dedicated CPU and memory resources, which guarantees access to those resources. SR-IOV allows each guest domain to have direct access to the physical device, although in the case of SR-IOV, the physical device is itself virtualized, and it is possible to have some resource contention, but no more than if the workloads were running natively on the same shared device.
 - » **Efficiency:** Oracle VM Server for SPARC technology allows the guest domains to make use of CPU, memory, and I/O resources directly without any virtualization overhead. This model also requires the allocation of resources to two service domains, rather than one in the previous model. In summary, this model almost completely removes the virtualization resource overhead, but some hypervisor overhead is still present.
 - » **Availability:** In contrast to the previous model, these domains are no longer fully reliant on the domain providing SR-IOV services, so both the resilient I/O domains would need to fail for guest domains to be interrupted. There are still failure scenarios that could cause the whole server to fail, causing all domains to fail. The guests I/O should be configured between the resilient I/O domains to provide redundant paths to networking and disk resources, so failures will have no impact to the guest domain.
 - » **Serviceability:** High serviceability is one of the main benefits of the resilient I/O domain approach in that the each I/O domain can be serviced in turn without affecting the running guests. The guests themselves can be independently serviced.
 - » **Flexibility:** SR-IOV domains are highly flexible. They can be dynamically resized, created, or destroyed relatively easily. They are slightly less flexible than guest domains due to the direct mapping of physical I/O to the domains.
- Agility:** Domains using non-virtualized I/O cannot be live-migrated, but they can be cold-migrated to other SPARC servers. This provides a capability to migrate workloads with relatively low service disruption. The target server must provide the same underlying I/O infrastructure, which makes this process slightly more complicated than with the Guest and Redundant Guest Domain models.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	Very High	High	Very High	High	Medium	Medium

Figure 5. Summary of the seven characteristics for the SR-IOV or Direct I/O Domains model

Guest Root Domains Model

Domains of this type are configured with PCIe root complexes that are directly assigned, giving them direct ownership of their I/O. This is the operationally simplest model because there is no need to create multiple virtual disk and network services. These guests run at a bare-metal performance level and are fully independent of each other. The number of root complexes and PCIe slots available limits the number of domains of this type that can be created. This model is ideal for environments where a small number of highly performant and independent domains is required.

Figure 6 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** Each guest root domain acts like a physical server with its own version of the operating system. It should be noted that one of the guest root domains will also have the role of the control domain and will, therefore, have the capability to manipulate the physical configuration of all other guest root domains on that server.
- » **Resource isolation:** Guest root domains have dedicated CPU, memory, and I/O resources, which guarantees access to those resources. Each guest root domain is completely isolated from the others. It is expected that each guest root domain will have access to boot devices and networking resources based on the I/O that has been assigned to it.
- » **Efficiency:** This model offers the highest level of efficiency because all the domains will operate at a bare-metal performance level, and there is no requirement for resources to be allocated to additional service domains. This model has zero hypervisor and virtualization resource overhead.
- » **Availability:** Because the guest root domain is fully independent from all the other domains, its level of availability is similar to that of a physical server, and the levels of redundancy are dependent on the underlying physical configuration, such as redundant I/O cards.
- » **Serviceability:** Each guest root domain within a server can be independently serviced without affecting any of the other domains. They are also capable of the usual hot swapping for I/O components.
- » **Flexibility:** The guest root domains are quite flexible. Their CPU and memory allocations can be dynamically resized, and they can be created or destroyed relatively easily. However, because they require I/O to be physically assigned to them at the root complex level, they are usually operated with a relatively static I/O configuration.
- » **Agility:** Guest root domains, like PDOMs and physical servers, cannot be migrated between systems.


Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
Very High	Very High	Extremely High	Very High	High	High	Low

Figure 6. Summary of the seven characteristics for the Guest Root Domains model

Oracle Solaris Zones

Oracle Solaris includes a built-in virtualization capability called Oracle Solaris Zones, which allows users to isolate software applications and services using flexible, software-defined boundaries. Unlike hypervisor-based virtualization, Oracle Solaris Zones technology provides OS-level virtualization, which gives the appearance of multiple OS instances rather than multiple physical machines. Oracle Solaris Zones technology enables the creation of many private execution environments from a single instance of the operating system, with full resource management of the overall environment and the individual zones. For Oracle software licensing purposes, Oracle Solaris Zones that are configured as capped or dedicated CPUs are considered to be hard partitions.

The nature of OS virtualization means that Oracle Solaris Zones technology provides very low-overhead, low-latency environments. This makes it possible to create hundreds, or even thousands, of zones on a single system. Full integration with Oracle Solaris ZFS and network virtualization provides low execution and storage overhead for those areas as well, which can be a problem area for other VM implementations. Oracle Solaris Zones technology enables close to a bare-metal performance level for I/O, making zones an excellent match for outstanding I/O performance.



Oracle Solaris 11 provides a fully virtualized networking layer. An entire data center network topology can be created within a single OS instance using virtualized networks, routers, firewalls, and network interface cards (NICs). These virtualized network components come with high observability, security, flexibility, and resource management. This provides great flexibility while also reducing costs by eliminating the need for some physical networking hardware. The networking virtualization software supports quality of service (QoS), which means that appropriate bandwidth can be reserved for key applications.

Oracle Solaris Zones technology also provides the ability to run older Oracle Solaris versions within zones. This capability is called *branded zones*. When running an Oracle Solaris 10 global zone, it is possible to run Oracle Solaris 8 and Oracle Solaris 9 zones within it. This allows legacy applications to be easily consolidated onto a more modern platform. Additionally, Oracle Solaris 10 workloads can take advantage of the network virtualization features of Oracle Solaris 11 by running Oracle Solaris 10 zones on top of an Oracle Solaris 11 global zone.

Oracle Solaris 11 allows Immutable Zones, which provides an additional level of security, because it makes the zones read-only and, thus, immune to malicious configuration changes or root-kit type of attacks.

The Kernel Zones feature in Oracle Solaris 11.2 provides a new type of zone that inherits all the benefits of OS-level virtualization with the additional benefit of running a unique kernel. This provides additional levels of isolation, specifically the ability to run different updates of Oracle Solaris within a global zone.

Oracle Solaris Zones are also integrated with Oracle Solaris DTrace, the Oracle Solaris feature that provides dynamic instrumentation and tracing for both application and kernel activities. For example, administrators can use DTrace to examine Java application performance throughout the software stack. It provides visibility both within Oracle Solaris Zones and in the global zone, making it easy for administrators to identify and eliminate bottlenecks and optimize performance.

It should be noted that a given global zone (the underlying Oracle Solaris instance running on the LDom/PDom/server) can simultaneously host any of the zone types listed below. This allows an appropriate zone type to be selected based on the workload requirements.

Further information about Oracle Solaris Zones can be found on the [Oracle Solaris 11 Virtualization Technology page](#).

Native Branded Zones

The term *native branded zone* is used in this document to refer to a zone that is running the same version of Oracle Solaris as the underlying global zone. It makes direct use of the kernel running in the global zone and provides the lowest overhead, because the performance of an application within a zone is no different from its performance running directly in the global zone. This type of zone provides a private execution environment for an application with the security and performance isolation that this environment provides.

The shared nature of the underlying kernel means that when the global zone is updated, all the zones of this type will also be updated, which adds some operational complexity as well as restricts the ability to run multiple OS update levels at the same time within a single system. This is usually mitigated by using scalable application architectures where these outages have no material effect, or by migrating some or all of the zones from the server prior to updating.

Figure 7 and the following list describe how this type of zone addresses the seven characteristics outlined earlier:

- » **Resource isolation:** Oracle Solaris Zones provide a wide range of extremely flexible resource controls. For CPU resource control, it is possible to specify dedicated CPUs, use capped CPUs, or use the Fair Share Scheduler to “share” the available CPUs among the zones using a weighted system. It is also possible to limit the use of memory in the zones using `rcap`. Network usage can be controlled using bandwidth settings for each of the links.

- » **Security isolation:** Oracle Solaris Zones offer an extremely high level of security because each zone is administratively separate from the others, and they can be made immutable if necessary.
- » **Efficiency:** Oracle Solaris Zones are extremely lightweight and provide near-zero virtualization overhead.
- » **Availability:** An Oracle Solaris Zone is as available as the underlying Oracle Solaris instance running in the global zone. On systems where many zones will be deployed, it is usual to configure the underlying system with high levels of RAS and redundant I/O. However, all zones are completely dependent on the underlying global zone, and if this fails, so do all the zones it is hosting.
- » **Serviceability:** Oracle Solaris Zones of this type are updated at the same time as the global zone. It is possible to migrate zones between global zones running the same OS version.
- » **Flexibility:** Oracle Solaris Zones are highly flexible. They can be dynamically resized, created, or destroyed extremely easily. Oracle Solaris Zones can be started and stopped rapidly because they are simply a set of processes on an already instantiated OS.
- » **Agility:** Oracle Solaris Zones, in this model, can be cold-migrated to other global zones running on SPARC servers. This provides the capability to migrate workloads with relatively low service disruption. The reboot time of a zone is significantly quicker than that of a physical server or a VM.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	High	Very High	Medium	Low	Very High	Medium

Figure 7. Summary of the seven characteristics for native branded zones

Kernel Zones

New to Oracle Solaris 11.2, kernel zones run their own unique kernel. This provides the capability to run different updates of Oracle Solaris 11.2 on the system, and it allows the kernel zones to be updated independently.

Figure 8 and the following list describe how this type of zone addresses the seven characteristics outlined earlier:

- » **Resource isolation:** Kernel zones provide the same resource allocation features of standard zones, with the added capability of dedicated memory allocation to each zone.
- » **Security isolation:** Oracle Solaris Zones offer an extremely high level of security, because each zone is administratively separate from the others.
- » **Efficiency:** Kernel zones are lightweight and provide near-zero virtualization overhead. Each kernel zone runs its own kernel and will, therefore, require more memory per instance than the traditional zones. This creates an additional overhead compared to traditional zones.
- » **Availability:** A kernel zone is as available as the underlying Oracle Solaris instance running in the global zone. On systems where many zones will be deployed, it is usual to configure the underlying system with high levels of RAS and redundant I/O. However, all zones are completely dependent on the underlying global zone, and if this fails, so do all the zones it is hosting.
- » **Serviceability:** Kernel zones can be updated independently of the underlying global zone. Of course, when the global zone is rebooted after being updated, this will still cause the kernel zones to be rebooted as well. It is possible to migrate zones between global zones running the same OS version, and kernel zones provide a warm-migration feature where a zone can be suspended on one server and resumed on another.
- » **Flexibility:** Oracle Solaris Zones are highly flexible. They can be dynamically resized, created, or destroyed extremely easily. Oracle Solaris Zones can be started and stopped extremely rapidly, because they are simply a set of processes on an already instantiated OS.

- » **Agility:** Oracle Solaris Zones in this model can be warm-migrated to other global zones running on SPARC servers. This provides a capability to migrate workloads with relatively low service disruption. The reboot time of a zone is significantly quicker than that of a physical server or a VM.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	Very High	High	Medium	Medium	Very High	High

Figure 8. Summary of the seven characteristics for kernel zones

Non-Native Branded Zones

The term *non-native branded zone* is used to define zones that run an older Oracle Solaris version than the version run in the global zone. For example, Oracle Solaris 10 allows Oracle Solaris 8 and Oracle Solaris 9 branded zones to be run, and Oracle Solaris 11 allows Oracle Solaris 10 branded zones to be run. In other respects, these types of zones share the same characteristics as traditional zones.

Figure 9 and the following list describe how this type of zone addresses the seven characteristics outlined earlier:

- » **Resource isolation:** Oracle Solaris Zones provide a wide range of extremely flexible resource controls. For CPU resource control, it is possible to specify dedicated CPUs, use capped CPUs, or use the Fair Share Scheduler to “share” the available CPUs among the zones using a weighted system. It is also possible to limit the use of memory in the zones using `rcap`. Network usage can be controlled using bandwidth settings for each of the links.
- » **Security isolation:** Oracle Solaris Zones offer an extremely high level of security, because each zone is administratively separate from the others.
- » **Efficiency:** Oracle Solaris Zones are extremely lightweight and provide near-zero virtualization overhead.
- » **Availability:** A zone is as available as the underlying Oracle Solaris instance running in the global zone. On systems where many zones will be deployed, it is usual to configure the underlying system with high levels of RAS and redundant I/O. However, all zones are completely dependent on the underlying global zone, and if this fails, so do all the zones it is hosting.
- » **Serviceability:** Zones of this type are updated at the same time that the global zone is updated. It is possible to migrate zones between global zones running the same OS version.
- » **Flexibility:** Oracle Solaris Zones are highly flexible. They can be dynamically resized, created, or destroyed extremely easily. Oracle Solaris Zones can be started and stopped extremely rapidly, because they are simply a set of processes on an already instantiated OS.
- » **Agility:** Zones in this model can be cold-migrated to other global zones running on SPARC servers. This provides a capability to migrate workloads with relatively low service disruption. The reboot time of a zone is significantly quicker than that of a physical server or a VM.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	High	Very High	Medium	Low	Very High	Medium


Figure 9. Summary of the seven characteristics for non-native branded zones

Combining Virtualization Technologies

As can be seen in Table 1, no single virtualization technology has the highest score for every single characteristic. This is simply because many of these characteristics are mutually exclusive.

TABLE 1. SUMMARY OF THE SEVEN CHARACTERISTICS FOR ALL VIRTUALIZATION TECHNOLOGIES

	Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
Full system	Extremely High	Extremely High	Extremely High	Extremely High	Extremely High	Low	Low
PDoms	Very High	Very High	Extremely High	Extremely High	Extremely High	Low	Low
Oracle VM Guest Root Domains model	Very High	Very High	Extremely High	Very High	High	High	Low
Oracle VM Redundant Guest Domains model	High	Very High	Medium	Very High	High	High	High
Oracle VM SR-IOV or Direct I/O Domains model	High	Very High	High	Medium	Low	Medium	Medium
Oracle VM Redundant SR-IOV Domains model	High	Very High	High	Very High	High	Medium	Medium
Oracle VM Guest Domains model	High	Very High	Medium	Medium	Medium	High	High
Oracle Solaris kernel zones	High	Very High	High	Medium	Medium	Very High	High
Oracle Solaris Zones	High	High	Very High	Medium	Low	Very High	Medium
Oracle Solaris branded zones	High	High	Very High	Medium	Low	Very High	Medium



By observing the characteristics of each of the different technologies, it should be clear that adopting a layered approach to consolidation by combining two or more of the Oracle virtualization technologies can create solutions that satisfy significantly more of the requirements than a single virtualization technology can satisfy.

For the purposes of this section, a physical server and a PDom are treated as essentially equivalent. The PDom configuration and sizing discussion is applicable only to the SPARC M7 and SPARC M6-32 servers, and the choice of the number and size of the PDom is mainly based on the workload sizing and serviceability requirements. In a 16-socket SPARC M7-16 server, you could choose to have one x16 socket PDom; two x8 socket PDom; four x4 socket PDom; or two x4 socket PDom and one x8 socket PDom. In the case of the SPARC M7-8 server, there is a factory-configured choice of a single PDom from 2 to 8 sockets, or a pair of PDom, each with 2 to 4 sockets each.

In general, given that there is no virtualization overhead associated with PDom, the only considerations for sizing is weighing up the serviceability and complexity requirements. Once the PDom have been selected, they can then be treated in the same way as independent physical servers.

The next step in defining the consolidation architecture becomes selecting the most appropriate way of deploying the workloads within the PDom or SPARC servers using LDom, Oracle Solaris Zones, or a combination of the two.

If we had 64 workloads that needed to be deployed, this could be achieved in a number of ways: At one extreme, we could configure 64 zones within a single LDom and at the other extreme, we could configure 64 LDom with one workload per LDom.

The 64-zones solution running on a single global OS instance in one large LDom provides the highest efficiency and a significantly reduced number of entities to be configured and managed. However, the high number of zones might have a serviceability impact and might not provide sufficient administrative, security, and resource isolation for the workloads. In the case where 64 LDom are deployed, each LDom will require its own boot disks and redundant virtual networks, and the LDom will need to be updated and managed as 64 independent servers, which has an impact on the operational complexity of the infrastructure.

The optimum choice is most likely to be somewhere in between.

The following section outlines a subset of the potentially useful combinations, and describes how the combined effect meets different workload requirements. These examples are by no means exclusive, and there are many combinations, as well as hybrid combinations, that could easily be deployed.

Redundant Guest Domains and Oracle Solaris Zones

This combination of virtualization technologies shown in Figure 10 allows a reduction in the number of LDom that need to be created by virtualizing each LDom at the OS level using Oracle Solaris Zones to allow multiple isolated workloads to be run per LDom. This achieves a simplified LDom configuration and a reduction in operational complexity, while still maintaining the main isolation benefits of LDom as well as the ability to live-migrate whole LDom between servers. Any of the three zone types described earlier can be deployed in this model, with the choice of zone type dependent on the workload requirements.

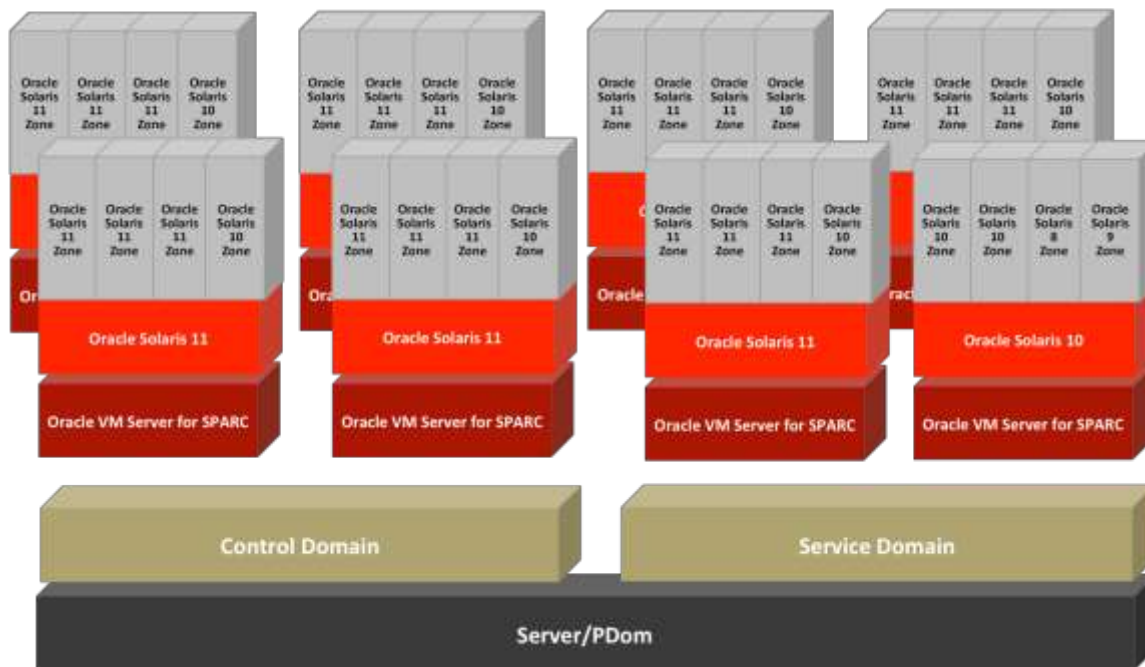


Figure 10. Example configuration that combines redundant guest domains and Oracle Solaris Zones

Figure 11 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** This model has two levels of security isolation. At the LDom level, there is high separation between the LDom, and there is a further level of isolation between the zones running inside the LDom. It is typical to group zones within the same security realm within the same LDom.
- » **Resource isolation:** Each guest domain has dedicated CPU and memory resources, which guarantees access to those resources as before. However, these resources can be further subdivided and shared in a more dynamic way among the zones running within the LDom.
- » **Efficiency:** This model is more efficient than the purely Redundant Guest Domains model simply because a smaller number of guest domains need to be created due to the use of zones as the additional level of workload isolation within the domains.
- » **Availability:** This model maintains the availability of the domains in terms of their ability to survive the loss of one of the domains providing I/O services.
- » **Serviceability:** High levels of serviceability is one of the main benefits of the multiple service domain approach in that the control and service domains can be serviced without affecting the running guests.
- » **Flexibility:** Guest domains are highly flexible. They can be dynamically created, resized, or destroyed relatively easily. The additional benefit of zones within the LDom is a finer-grained level of flexibility, because zones can also be dynamically created, resized, or destroyed.

- » **Agility:** Guest domains can be live-migrated or cold-migrated to other SPARC servers. This provides a capability to migrate workloads with little or no service disruption. Alternatively, single zones can also be moved between guest domains as required.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
High	Very High	High	Very High	High	Very High	High

Figure 11. Summary of the seven characteristics for a configuration that combines redundant guest domains and Oracle Solaris Zones

Guest Root Domains and Oracle Solaris Zones

The model shown in Figure 12 combines the zero virtualization overhead and high isolation of guest root domains with the flexibility and agility of Oracle Solaris Zones to provide the highest possible efficiency with the lowest level of overhead. The use of kernel zones for certain workloads allows additional isolation at the zone level, particularly in the area of memory resource management, and the ability to warm-migrate those zone workloads. This model typically has a smaller number of LDomS running a larger number of zones.

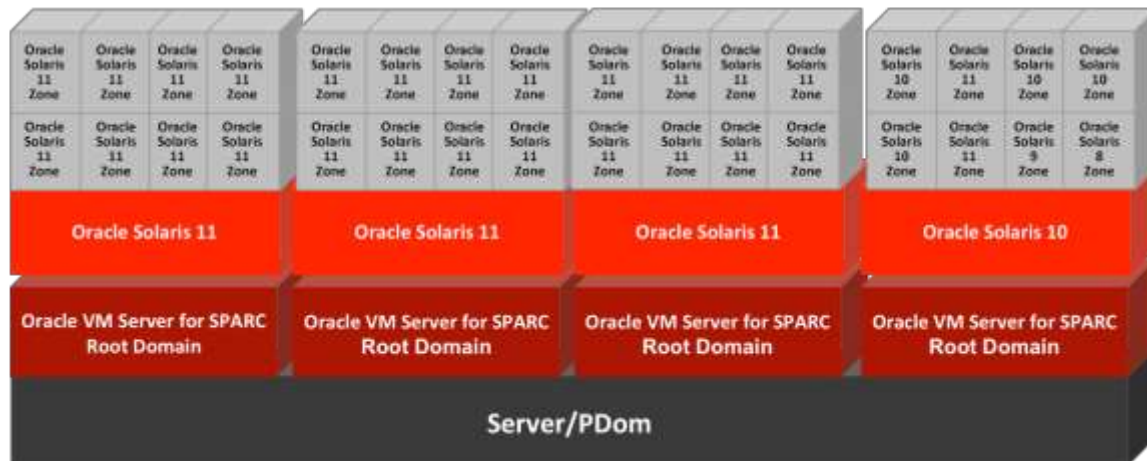


Figure 12. Example configuration that combines guest root domains and Oracle Solaris Zones

Figure 13 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** This model also has two levels of security isolation, although compared to the previous example, the guest root domain is running on dedicated hardware. At the LDom level, there is high separation between the LDomS, and there is a further level of isolation between the zones running inside the LDomS. It is typical to group zones within the same security realm within the same LDom.
- » **Resource isolation:** Each guest root LDom has dedicated CPU, memory, and I/O resources, which guarantees access to those resources as before. However, these resources can be further subdivided and shared in a more dynamic way among the zones running within the LDomS.

- » **Efficiency:** This model is the most efficient because there is no hypervisor overhead, and there is also no virtualization overhead. It also provides the most operationally simple model, because it provides the lowest number of entities that need to be managed. In this way, it avoids all three types of virtualization tax.
- » **Availability:** Each guest root LDom is a fully independent domain and if they are configured with redundant I/O, they are more available than the redundant guest domain model described earlier. The zones are, of course, still fully dependent on the underlying global zone and the guest root domain.
- » **Serviceability:** In this model, the root domains themselves have a level of serviceability due to redundant I/O connections; however, in many cases, the guest root domain will usually be brought out of service for work to be done. The use of zones allows these workloads to be migrated more granularly.
- » **Flexibility:** Guest LDoms are highly flexible. They can be dynamically resized, created, or destroyed relatively easily. The additional benefit of zones within the LDoms is a finer-grained level of flexibility.
- » **Agility:** Root domains cannot themselves be migrated between servers; however, the addition of Oracle Solaris Zones adds the capability for these workloads to be easily migrated off the root guest domain. Kernel zones provide the capability to warm-migrate through the use of suspend and resume functionality.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
Very High	Very High	Extremely High	Very High	High	Very High	High

Figure 13. Summary of the seven characteristics for a configuration that combines guest root domains and Oracle Solaris Zones

Root Domains and SR-IOV Domains

While this mechanism uses the same technology for both layers (Oracle VM Server for SPARC), it provides yet another useful model. It is similar to the Guest Root Domains with Oracle Solaris Zones model, but it uses the root domains purely as SR-IOV servers for a large number of SR-IOV domain guests. A root domain is distinct from a guest root domain because no application workloads actually run within a root domain. The root domains are sized with a large amount of I/O, but with modest CPU and memory allocation. While it is possible to also layer Oracle Solaris Zones on top of these guests (as shown in Figure 12), for simplicity, we will only consider a two-layer model.

The SR-IOV domain's dependence on the controlling domain means that it is important to create multiple failure domains within a single server so that the effect of the failure of a service domain is limited only to the guests it is servicing. As an additional refinement, the root domains could be configured in pairs providing resilient I/O services to the guest domains.

This model provides the highest level of isolation while still maintaining excellent performance levels by allowing the use of a relatively high number of SR-IOV domains within a single system. Its availability characteristics are similar to the previous model, because in both cases, there is a dependency on the root domain (either hosting zones or hosting SR-IOV guests).

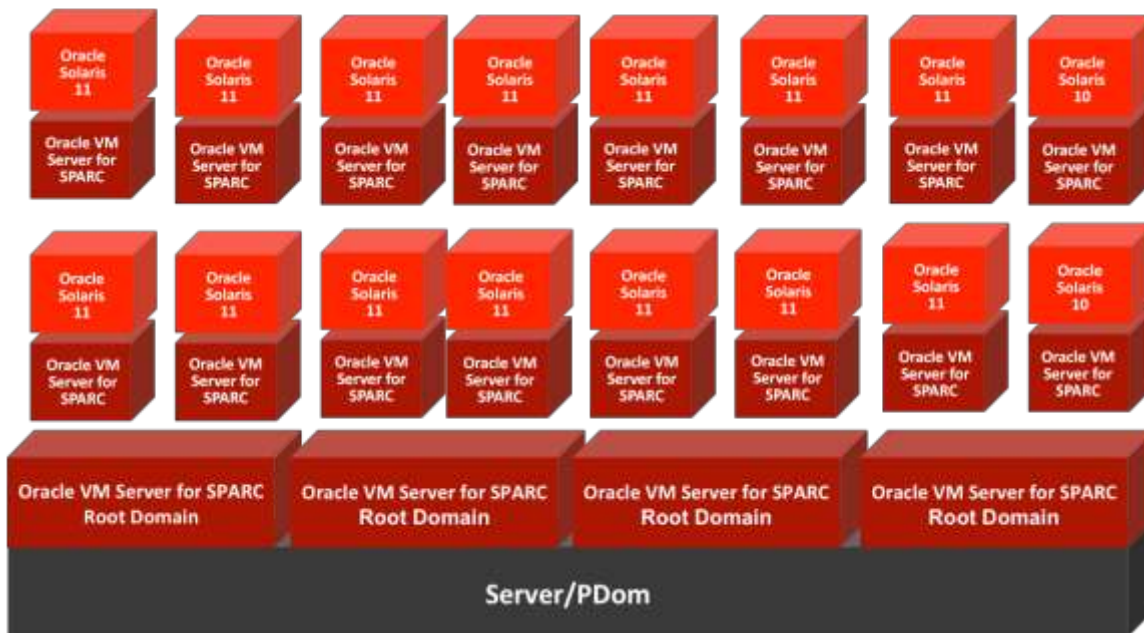


Figure 14. Example configuration that combines root domains and SR-IOV guests

Figure 15 and the following list describe how this model addresses the seven characteristics outlined earlier:

- » **Security isolation:** Each guest domain acts like a physical server with its own version of the operating system. It is serviced by a single root domain, and SR-IOV guest domains within the same security realm could be serviced by the same root domain.
- » **Resource isolation:** Each guest domain has dedicated CPU and memory resources, which guarantees access to those resources. The use of SR-IOV allows each guest domain to have direct access to the physical device, which directly supports multiple virtual connections. It is possible to have some resource contention, but no more than if the workloads were running natively on the same shared device.
- » **Efficiency:** Oracle VM Server for SPARC technology allows the guest domains to make use of CPU, memory, and I/O resources directly without any virtualization overhead. However, this model requires the allocation of resources to the service domain, although this should require fewer resources than the models that use virtualized I/O. In summary, this model completely reduces the virtualization resource overhead, but the hypervisor overhead is still present.
- » **Availability:** The use of SR-IOV places a dependence on the domain that owns the physical device. If this domain has an unplanned failure, the failure will also affect the guests owning the dependent devices and cause undefined consequences, usually a panic. Introducing multiple domains providing SR-IOV services to groups of guest domains reduces the impact of single failures.
- » **Serviceability:** If the domain that owns the physical device requires servicing, the guest domains will also need to have an outage or be migrated to other servers. However, the guest domains can be serviced without affecting any other guest domains, and root domains can be serviced and affect only the guests they are servicing.
- » **Flexibility:** Guest domains are highly flexible. They can be dynamically resized, created, or destroyed relatively easily.

» **Agility:** Guest domains with non-virtualized I/O can be cold-migrated to other SPARC servers. This provides a capability to migrate workloads with relatively low service disruption.

Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
Very High	Very High	High	High	High	Very High	Medium

Figure 15. Summary of the seven characteristics for a configuration that combines root domains and SR-IOV domains

Hybrid Combination of All Oracle Virtualization Technologies

The flexibility of Oracle’s virtualization technologies is that they are not mutually exclusive. Namely, it is always possible to run any type of zone on any Oracle Solaris instance, and the Oracle VM Server for SPARC technology allows each root domain within the system to perform different functions, which provides maximum flexibility to deal with a wide variety of workload types.

Figure 16 shows the first two root domains as a redundant pair of domains providing I/O services for the four redundant guest domains (A), the third domain is a root domain providing SR-IOV services to the four guest SR-IOV domains (B), and the final and fourth root domain is using the purely Oracle Solaris Zones model (C).

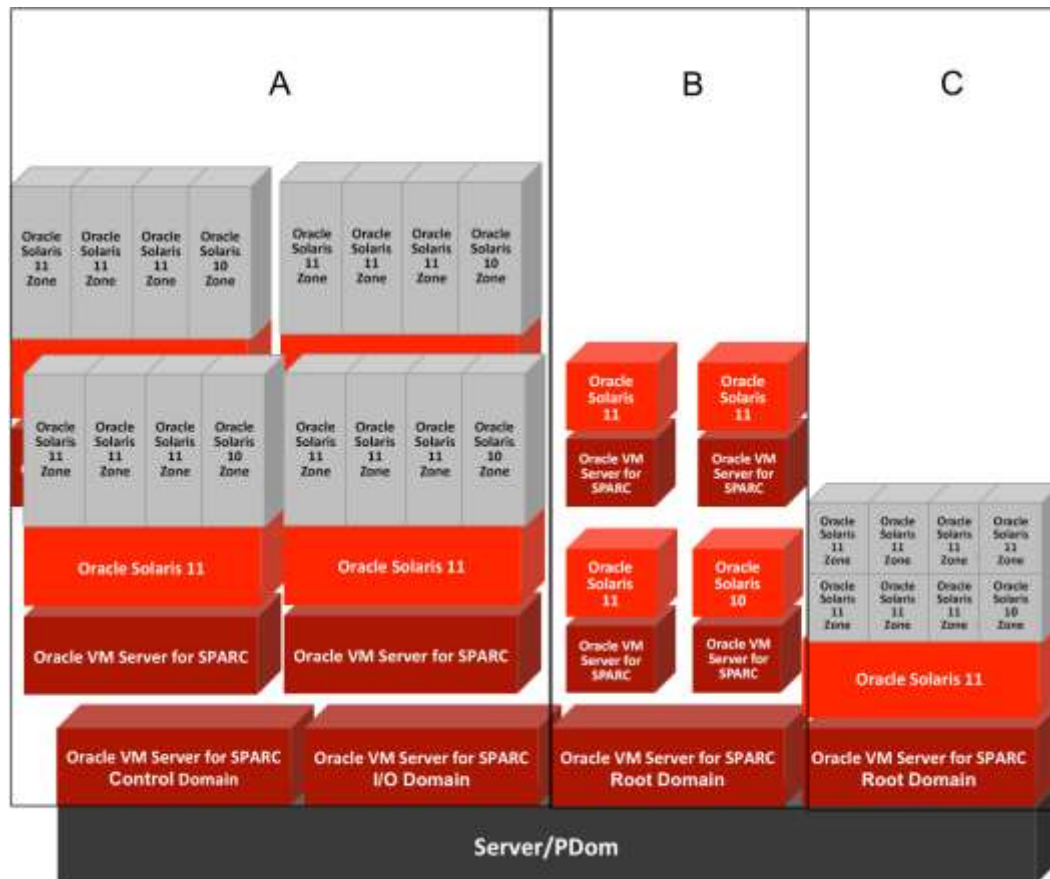


Figure 16. Example hybrid configuration that combines all Oracle virtualization technologies

As can be seen, it is possible to incorporate each of the models within a single server/PDom, and their virtualization characteristics are identical to the combined models outlined earlier.

Summary of Characteristics for Combined Virtualization Technologies

Each of the combined approaches allows higher scores across the range of virtualization characteristics compared to using only one of the technologies. As shown in Table 2, by adopting a layered approach to virtualization, it is possible to build architectures that meet a wider range of workload requirements.

TABLE 2. SUMMARY OF THE SEVEN CHARACTERISTICS FOR COMBINED VIRTUALIZATION MODELS

	Security Isolation	Resource Isolation	Efficiency	Availability	Serviceability	Flexibility	Agility
Redundant Guest Domains and Oracle Solaris Zones	High	Very High	High	Very High	High	Very High	High
Guest Root Domains and Oracle Solaris Zones	Very High	Very High	Extremely High	Very High	High	Very High	High
Root Domains and SR-IOV Domains	Very High	Very High	High	High	High	Very High	Medium



Conclusion

This paper has highlighted the characteristics of each of Oracle's virtualization technologies, and shown how the technologies can be combined to provide architectures that more closely match typical workload requirements instead of deploying a single virtualization technology.

One size does not fit all, and the key to successful consolidation is to create a flexible infrastructure that can easily accommodate all workload types.

It is also important not to lose sight of the original reasons for consolidating in the first place, which is to cut costs by

- » Reducing the size of the hardware and software infrastructure
- » Improving efficiency
- » Reducing operational complexity

Combining the Oracle virtualization technologies of physical domains, Oracle VM Server for SPARC, and the various types of Oracle Solaris Zones enables an agile and flexible architecture to be designed to meet those goals.

About Oracle Elite Engineering Exchange

Oracle Elite Engineering Exchange (Oracle EEE) is a cross-functional global organization consisting of Oracle's elite sales consultants (SCs) and systems engineers (Product Engineering). Oracle EEE connects systems engineers directly to the top experts in the field through joint collaboration, which enables bidirectional communication about customer and market trends and deep insight into the technology directions of future-generation products. Oracle EEE brings real-world customer experiences directly to systems engineers and engineering technical details and insights to sales consultants, both of which enable better solutions to meet the changing needs of Oracle's customers.







Oracle Corporation, World Headquarters

500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries

Phone: +1.650.506.7000
Fax: +1.650.506.7200

CONNECT WITH US

-  blogs.oracle.com/oracle
-  facebook.com/oracle
-  twitter.com/oracle
-  oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2015, Oracle and/or its affiliates. All rights reserved. This document is provided *for* information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0615

Consolidation Using Oracle's SPARC Virtualization Technologies
October 2015
Author: Michael Ramchand, Peter Wilson, and Martien Ouwens