WHITE PAPER

Cost-Per-Application – Definition and Use

vmware®

What is Cost-Per-Application and why it is the key metric to compare upfront costs of virtualization solutions?

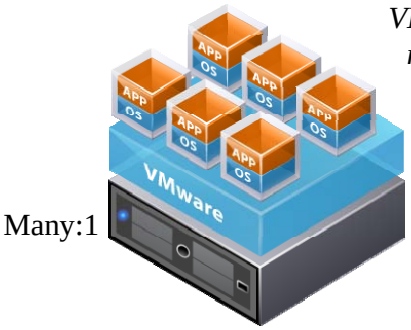
Virtualization changed the way applications are deployed, making it possible to run multiple applications independently on the same sever.

Physical World

- **1:1 relationship** between applications and hardware
- Relevant cost metric = **upfront licensing**

Virtual World

VM density matters!



- **Many:1** relationship between applications and hardware
- Relevant cost metric = **cost per application**

The number of applications that a virtualization solution allows to run per host has direct implications on the overall cost of its deployment. A higher number of applications per virtualization host, or VM density, translates into:

- Lower infrastructure cost (less servers, networking and storage)
- Lower power and cooling costs
- Lower data center real estate cost
- Lower guest operating systems cost (Windows Server 2008 Datacenter Edition is licensed by CPU, so less servers means less CPUs to license)

In evaluating the cost of a virtualization solution, it is essential to use a metric that not only looks at the cost of licensing the virtualization software, but also takes into account other infrastructure components such as servers, networking, storage, power, real estate and guest operating systems licensing.

In addition, because centralized management is a fundamental component of any virtualization deployment, hardware and software costs associated with related management products, such as VMware vCenter or Microsoft System Center or Citrix Essentials need to be included as well.

VMware defines **cost-per-application** as the sum of all these cost items divided by the number of applications virtualized

	Hardware Cost (virtualization hosts, management servers, networking, storage)
+	Power and Cooling Cost
+	Datacenter Real Estate Cost
+	Guest Operating System Software Cost
+	Virtualization Software Cost
+	Virtualization Management Software Cost
=	Total Cost of Deployment
/	Divided by Number of Applications Virtualized
=	Cost per Application

Ultimately, companies care about how much it will cost to run the *total* set of their business applications and how well they are utilizing their infrastructure. Therefore, in a virtual IT environment, **cost-per-application** is the key metric when comparing costs.

Why VMware makes it possible to run more applications per server?

Due to its advanced technologies and architecture, VMware Infrastructure 3 makes it possible to successfully run more applications per host (physical server) and consequently achieve a lower cost per application than commodity virtualization offerings. The most significant reasons why VMware Infrastructure can enable a higher VM density per host include the following:

1) **Memory Oversubscription** – VMware ESX makes more efficient use of physical RAM by reclaiming unused physical memory allocated to particular VMs (think of it as de-duplication for memory) and consolidating identical memory pages among VMs on a host. Both functions let ESX oversubscribe memory on a server with minimal impact to performance. This memory oversubscription is no different than oversubscribing the CPU and network on a virtual host. Commodity offerings, such as Microsoft Hyper-V, Citrix XenServer or Virtual Iron, lock the memory assigned to a VM so that no other VM can use it, regardless of whether the memory is being used or not. Locking memory assigned to a VM leads to very inefficient use of physical RAM and lowers the number of virtual machines that can be successfully hosted on a single virtualized server.

2) **Direct Driver Model** – VMware optimizes device drivers for virtualization and places them directly in the ESX hypervisor. By giving VMs direct and fast access to the hardware VMware ESX can achieve very high I/O throughput and can handle the I/O requirements for more VMs simultaneously requesting hardware resources. Other hypervisors, such as Hyper-V or Xen-based ones, use an indirect driver model that leverages the generic device drivers, located in a separate partition (Parent Partition or Dom0) of the virtualization software stack that contains an instance of a general purpose operating system (Windows Server 2008 or Linux). This “indirect” architecture single threads the I/O requests from all VMs through a single channel into the Parent Partition (or Dom0) instance of Windows Server (or Linux), forcing the VMs to fight over resources in the operating system. This creates an I/O bottleneck that reduces the I/O throughput commodity hypervisors can achieve; as more VMs are run on the same host, the I/O bottleneck gets worse. The indirect driver model also makes every VM reliant on third-party device drivers that have achieved only basic WHQL certification and are known for their instability.

3) **High Performance "Gang" Scheduler** – VMware designed and optimized the ESX process scheduler to handle the resource requirements of many heterogeneous VMs. Commodity hypervisors use the generic process scheduler of general purpose operating systems. These were not designed for virtualization nor do they meet the special needs of multiple VM workloads. Generic process schedulers, like the Windows scheduler used with Hyper-V, were designed to fulfill different use cases, such as providing an effective user interface experience. Because of its reliance on the Windows scheduler, Hyper-V gets hit by performance limits at levels far lower than those of VMware ESX and can run fewer VMs per host than VMware ESX. By comparison, the architecture of the ESX "Gang" scheduler is optimized for virtualization. It can dynamically account for the CPU and I/O needs of virtual machines by dynamically allocating more resources and larger processor timeslices to VMs.

4) **Support for Large Memory Pages and Nested Page Tables** – Large memory pages and nested page tables optimize memory access and can provide substantial performance benefits for mission critical, memory-intensive applications, such as an Oracle database. Large memory pages and nested page tables allow VMware solutions to achieve high levels of performance while consuming significantly fewer CPU resources. VMware ESX supports both features. Test results show that large memory pages and nested page tables can reduce CPU resource consumption by up to 15%. VMware ESX automatically redistributes the freed CPU power to other workloads, ultimately allowing higher VM density. On the other hand, Microsoft Hyper-V does not support these features and consequently forces users to allocate more hardware resources and run them at a lower VM density than VMware ESX.

5) **DRS with Resource Pools –** VMware Infrastructure can dynamically load balance VMs across a cluster so applications get required resources when they need them. This reallocation of resources happens automatically and without service discontinuity, based on service levels set by the application owners. DRS is essentially a "safety net" that lets administrators run individual servers at higher utilization levels while meeting service level agreements—even when spikes occur. DRS lets usage spikes that might overwhelm a single server be leveled across many servers in a cluster with no interruption to the end user. Microsoft Hyper-V does not have any comparable feature. The closest is its PRO Tips, but that is handicapped by its Quick Migration technology that forces downtime every time a virtual machine is migrated, resulting in unacceptable performance interruptions.

Bottom Line:

With these and other product advantages VMware customers report that they can achieve 50-70% higher VM density per host than with commodity virtualization offerings. This results in a 20-30% lower cost-per-application on average.

VMware, Inc. 3401 Hillview Ave Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com

vmware®