Clipper Notes™
VENDOR-AGNOSTIC EXPLANATIONS AND ADVICE
FOR THE INFORMATION TECHNOLOGY BUYER
Navigating Information Technology Horizons℠

New in 2007 Report #TCG2007029 February 28, 2007

Virtual Machines — Three Things to Consider + Three Ways to Use

Part 2 of a Multi-Part Series on Server Virtualization

Analyst: Anne MacFarland

Management Summary

Most of us own cars or trucks, or something in between. Some people own several, so that they can haul a dirt bike in the truck and visit clients in a luxury sedan. We're stuck being owners, caretakers, insurers, etc., of physical machinery. If you live close to work or your mass transit station, it is probably safe to say that your vehicle sits idle almost all of the time. If you divided the cost of your vehicle by the hours actually used in a year, the cost per hour used would be very high. If you used your vehicle irregularly, as you might if you lived in a city and used public transportation, you might conclude that it is a lot cheaper, and not terribly impractical, to have no personal vehicle. If you need a car, you could rent one from one of the many rental agencies. In a sense, this rental car becomes your "virtual vehicle"; you know that one will be available when you need it.

This virtual vehicle makes economic sense, especially when your needs vary. If, in winter, you need to go into the mountains for skiing, you could get a 4-wheel drive vehicle to get you there. This "container" matches your need. If all you need is to get to a meeting twenty miles away, a more modest "people's car", like a VW bug from ZipCar, might be the proper "container" to get you where you want to go (i.e., a good "virtual vehicle"). While it might be easier to keep a fleet of vehicles in your garage or driveway, it isn't a very efficient deployment of your assets.

These days there is much ado about virtual machines and containers. You need to know the difference and why they might make eminent sense for your computing environment, just as a properly-sized, temporarily-deployed vehicle makes sense if you live in the city. You can think of the **virtual machine as something that does what you want it to do when you need it to be done.** In server terms, this can be a two-second-long transaction execution or it can be an application that is always on and available. Both "applications" need to exist in an environment that makes execution possible, i.e., an instance of operating system. In the "old days" of not too long ago, each might sit within its own physical (scale-out) server. If you go back even further (to "the good old days"), both might have been operating within the same scale-up server. Now there are more options.

Some might say that the virtual machine is the container, while others might argue that the virtual machine hosts many containers or that the containers can have more than one virtual machine in it.[1] Some see the virtue of a virtual machine as the way it lets applications be deployed more quickly. There are different approaches for the use of virtual machines as infrastructure (this bulletin will discuss three), just as there are subtle differences in virtual machines (this bulletin will discuss three of these, too). These differences are worth exploring. Unlike that four-wheel-drive vehicle you might occasionally rent, virtual machines are something most IT environments will use in quantities that make these differences matter.

Here's the bottom line. **Virtual Machines are useful wherever you want an IT environment capable of delivering the virtual processing power to your applications.** What you should look for, and what you can expect, is the subject of this bulletin. For more details, please read on.

IN THIS ISSUE

[1] As was noted in Mike Kahn's "Server Virtualization Made Real", Part 1 of this multipart series of Clipper Notes, this gets even more confusing when "VM" is thought to represent Virtual Machine – or is it Virtual Memory? – or is it both? This is more than semantic variation. See this issue of Clipper Notes at http://www.clipper.com/research/TCG2007028.pdf.

The Basics of Virtual Machines

The driving force behind the current popularity of virtual machines is that of server consolidation. Energy is only the newest poster child for server consolidation, joining its more seasoned sibling the unmanageability of server sprawl. Doing more with less hardware makes sense. However, the programming that underlies most applications involves certain boorish assumptions about resources (it's-all-mine) that call for some form of containment to make co-location of applications work. Partitions, both hardware and software, are one way to address the problem, but they, alone, may not address it all. Management of a partitioned server environment becomes much more complex and just as skills-intensive, and deployment of additional applications on a server is hard to adequately test[2].

History

Virtual machines have been around for a long time, but it is when they became available on Intel servers that they became widely noticed. *Java Virtual Machines* were of interest mainly to Java developers and z/VM was limited to the mainframe community. The targeting of VMware found quick favor, first by developers for easy debugging, and then at datacenters for easy deployment and the ability to break the application-per-server quick drain on the budget. (For a quick overview of architectural approaches to Virtual Machines, see Exhibit 1 on the next page.)

Abstraction

Virtual machines add a layer of abstraction to the hardware and software stack that supports an application, and, as boundaries, are an opportunities to add additional features, or "intelligence." The abstraction can come in the form of emulation or translation of instructions, or a brokering of expectations between the application and the application. This sharply reduces work needed to meet the expectations - meticulously - that traditional configuration demands.

Hypervisor

Most, but not all, virtual machines products involve a resource monitor and manager called a *hypervisor*. It can be located, in the virtual machine software, in the host operating system or in the hardware's firmware – or some combination of both. The hypervisor monitors resource usage and brokers the integration with the underlying hardware instruction set. This brokering, allows multiple applications to share the resources of a processor. It eases deployment of applications on hardware and facilitates their complete removal. Additional container functionality can include authentication and other forms of security. Multiple layers of hypervisors give a greater granularity of control over the infrastructure than does a single element. The broader your use of virtual machines, the more this granularity matters.

Advanced Features

The ability to monitor resource use at a fine granularity lets more advanced virtual machines share unused resources with each other. Further evolution allows applications of lower priority to be limited in their memory use to allow high-priority applications the memory resources they need. The more customer-driven your business model, and the more stingy your technology budget becomes, the more relevant these advanced capabilities become.

Vendors and industry groups are all contributing to standards in the area of these guest environments called *VMs*, for as an important infrastructure enabler, these containers must be secure (as a point of control, Hypervisors are an obvious hacker target) and, in the end, sufficiently compatible to co-exist.

As components of your IT infrastructure, the decision that you make about virtualizing your servers is a matter of the long term. As they are software products, their evolution is rapid. Consider the benefits and limitations of virtual machines and their alternatives.

Are You Building Space Shuttles or Space Stations?

This is a question that must be addressed in order to frame your exploration of what virtual machines can do for you. Are you building the equivalent of space shuttles, or even the more lightweight drones, as an application deployment strategy? Or, are you building a more permanent, always evolving, space station – something that is more of a matter of architecture? It might seem that the issue of server consolidation is primarily a matter of deployment – **but virtualization changes things.** Where it once made sense to keep applications apart to localize failure, the fault-isolation and system resilience that virtual machines bring to infrastructure means that co-locating applications to enhance the speed of the workflows and business processes they support makes good sense, particularly when you are talking about hundreds of thousands of iterations of them each day.

There are advantages to each strategy, depending on your business and its immediate challenges. Which approach you take will make a difference in what features you seek.

[2] Virtual Machines and other container approaches are only one type of server virtualization. Virtualization of server hardware into large pools of capacity, or of their operating systems into something easier to manage, is another approach. Virtualization of operational utility services (scheduling, deploying, starting, stopping etc. across pools of resources) is often known as Grid. All these kinds of virtualization focus on the physicality or physical characteristics) required to make the physical assets seem like something different. In this tutorial, we will address the kinds of virtualization focused on the container.

The Inherent Pessimism of Space Shuttles

The way an organization will use virtual machines depends on its attitude towards infrastructure. *Are you going to assume a fairly high rate of failure and replacement?* This Zen - an *optimize-the-moment* kind of attitude - will optimize the ability to provision another unit of infrastructure quickly. It focuses on better using the servers that are the building blocks of physical infrastructure.

The Inherent Optimism of Space Stations

Are you more focused on the long-term support and optimization of your business processes? Are you trying to build some unique differentiator through your use of technology that your competitors might not be able to match? As scale-out architectures - like Web Servers - become less a matter of shoveling out pages, and more a matter of supporting not just the transactions of e-commerce but two-way interaction, video, and even social spaces, the demands for co-location and coordination of process components. Virtual machines are a tool of this approach as well, and, of course, there is a lot of overlap in how virtual machines are used for each strategy. The difference is whether you focus on *virtual machines as the deliverable*, or whether you think of *virtual machines as a tool* to create a larger, more integrated whole.

Three Key VM Qualities to Consider

(1) Overhead

All virtual machines – in fact, all forms of virtualization, work by proxying or translating requests from the guest to the host. This is what allows virtual machines to be rapidly deployed and/or moved. The proxying has a certain overhead. There are places where overhead is not a primary consideration (most desktops, some branch offices). There are other situations (many involve intense numerical calculations) where overhead cannot be tolerated. This is not a matter of openness or vendor allegiance, it is a matter of business requirement.

A virtual machine has its own needs for CPU cycles and memory. The more virtual machines you cram on a server, the more the inherent overhead of whatever virtual machine approach you have chosen will accumulate. If you have an approach that uses both host and guest operating systems, the cost of buying and/or supporting these operating systems will add costs as well. Remember that these costs are incurred per virtual machine. If you use many, they will become a significant factor.

(2) Granularity/Flexibility

The granularity[3] of the virtual machine can affect - directly - the extent to which you can consolidate your physical infrastructure while still supporting your applications. How many applications can be crammed on a processor depends on their use of resources and their profiles of activity, and your selection of *what* will be co-located *where*. If certain workloads work well the closer they are co-located, virtual machines can give them an environment where they can really hum. Depending on the applications that are to be virtualized, a business may need the expansive virtual machines that can span processors.

Sharing of resources[4] can boost optimization and assist the coordination of process. **How well it does the latter is often a function of what *sharing of resources* really means.** Sharing can mean the peaceful coexistence of *I've-got-mine* and *you've-got-yours* and we're not going to get any more unless we-kill-him-off, which reduces contention but does not foster process optimization. There can also be more abstemious allotments and a separate, brokered pool of resources that can be used by any process that needs it. In some products, the broker can find unused resources and reallocate them. The more you are trying to build differentiation through the use of virtual machines, the more theses advanced forms of sharing will be attractive to you.

(3) Manageability

Virtual Machines have the same management needs that their physical counterparts do. IT operations must be able to monitor them as it does other IT infrastructure elements. The ability to trap events and communicate them is a basic part of the current crop of virtual machines, but it is a good idea to make sure your virtual machine satisfies your organization's governance and auditing requirements. You should also check to see what kind of database the virtual machine software requires.

Take a careful look at how your environment is managed. If your operating systems and platforms are not equipped to recognize virtual machines, application faults that occur inside them become very hard to diagnose. The makers of virtual machines provide software to manage them, but it is best if these management tools integrate into your server and framework tools so that one management interface can see all that is needed.

[3] IBM's z/VM, which is more expensive than other virtual machines, can host many more virtual machines under one license and many more virtual machines on a single engine (processor complex), due to the coordinated nature of its virtualization. This is a long-term infrastructure element, and up-front costs are far from the only cost to be considered.

[4] Resources include processor cycles, memory, networks, and I/O. The solution will be the least common denominator of what both the hardware and the virtual machine software support.

Exhibit 1 – Processor Virtualization by Guest Containment

Basic Architectural Approaches

Specific vendor offerings, over time, are moving from the limitations of partitioning and high overhead of emulation towards the efficiency of chip-assisted approaches. Many offer more than one approach. Rather than considering quickly-outdated specific offerings, we list approaches.

- *Partitioning* (most vendors have multiple forms of partitioning) is one kind of guest containment, and it can form a part of some virtual machine solutions. The many kinds of partitions have different controls. They allocate processors in whole or parts by divvying out shares. If a workload is removed, the shares grow.
- The original virtual machines *emulated* the hardware. Emulation carries significant overhead, but it is still useful where a certain environment has to be supported for an application, or for developers wishing to port to different chip architectures.
- *Trapping and translation* of commands from the application by the guest operating system, which communicates with the hypervisor, is sometimes called *classic* or *pure* virtualization. This is more efficient than full emulation - but overhead still is a factor for computationally intensive operations.
- Some forms of virtualization require a *host operating system as well as guest operating systems* in addition to a hypervisor. This limits the number of virtual machines that can be co-located on a processor, and the ability of a virtual machine to span processors
- *Paravirtualization* works by modifying the guest operating system, removing the need for binary translation. This is more efficient than trap and translate, for the guest operating system can directly call the hypervisor. Paravrtualization works well with current versions of Linux, but less well with Windows and older Linux versions.
- *Hypervisor-assisted* virtual machines systems use an additional hypervisor element to mediate commands between the guest operating system and the underlying hardware – translating only those instructions that are otherwise difficult to broker.
- *Bare-metal or full virtualization* is still more efficient. There, the firmware does much of the virtualization, making it more efficient – but usually platform-specific. z/VM moved to an intelligent hypervisor arbitrating with firmware on a full set of resources (cycles, memory, network, and I/O) on the z9 processor in 2000. With Intel's *VT* and AMD's *Pacifica* support for virtualization (processor-only at this time), VMWare *ESX* offers a similar support for full virtualization.

Yes, all these approaches resemble the system coordination that is done on a larger scale in data center aggregations of boxes. The smaller scale cuts costs, and saves money and time.

Additional Considerations

Guest Operating System Cooperation

- Linux 2.6.20 supports KVM, which transforms the Linux Kernel into a Hypervisor by adding a new execution mode.
- Windows, at present, has inherent structures that require some translation or emulation.
- Mainframe operating systems, including z/VM, were developed to work together. In theory, many operating systems could be made to work as guests of virtual containers.

Host Operating System Cooperation

- IBM System z architecture actively co-operates with z/VM to virtualize CPU, memory, I/O, cryptographic, and networking resources.
- IBM *POWER* architecture supports virtualization via partitions.
- Intel *VT* and AMD *Pacifica* will facilitate virtualization by VMWare, XEN, Microsoft Virtual Server, and other products focused on those platforms.
- Sun's *SPARC* architecture has included virtualization for many years.

Different Uses of Virtual Machines

(1) Containers for Isolation: A Matter of Resilience

The most obvious use of virtual machines in business systems is to isolate the greed, corruption, illness, or death of an application inside a virtual machine from applications inside other virtual machines and from the host. This is the kid glove aspect of virtual machines, and it can foster the system resilience that today's business processes demand.[5] **The isolation of the container is the key asset.**

VMware has targeted the desktop as another venue where virtual machines can be useful. It has focused mostly on server-based thin-client deployments. Test environments are another use case – applications can be safely tested on real equipment (but in virtual isolation). Developers can set up multiple environments and safely test risky alternatives.

(2) Containers for Managing the Contents – To Make Deployment Easier

In scale out environments that must respond to bursts of demand, the ease of deployment is the primary virtue of a virtual machine. The ease of deployment, coupled with virtual machine's ability to maintain an inventory of application and operating system images as stored files[6] allows operations to focus on management of applications as assets, rather than burdens to be maintained.

The availability of products like VMWare's *VMotion* means that, with virtual machines, applications and even whole environments can be delivered to branch offices and other locations with no skill sets but a need for local functionality. Remote management capabilities are key to making this kind of use work. Should an environment malfunction, a replacement can be downloaded.

Containers for Proximity – A Matter of Process

In IT architectures, as in other matters of geography, location matters. If related application processes can be co-located, the business processes they jointly support will be enhanced. This kind of use of virtual machines rests firmly on the ability of a hypervisor to allocate and reallocate resources, and to promote sharing of both memory and common procedures. It is in this area that the System z's z/VM brings some unique assets to the job,[7] but any virtual machine can be used to co-locate applications that benefit from tight do-ordination.

The ability to reallocate resources between virtual machines, and the ability to throttle certain machines running lower priority workloads is useful. More important is the ability to support, with the hypervisor, the communications by which application elements fully support a business process. Together they make virtual machines, not just a matter of money-saving consolidation, quick recovery from failure, and application mobility, but an essential component of developing real-time business processes that span the traditional tiers based on physical IT assets.

First Steps to Take

Take a careful look at your existing environment and the capabilities in it that you cherish. Do you use 64-bit applications? Make sure they are supported as guest operating systems. Do you need to support symmetric multiprocessing in your guest environments? Make sure your choice of virtual machine environment can do that. What demands do your applications place on system resources, and you? Does that affect your consolidation plans? Make a list of your requirements, and check vendors both for their present capabilities and for the directions of their product roadmaps.

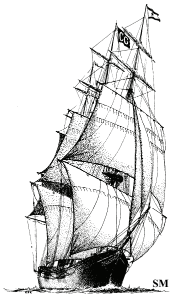
How many virtual machines you can on a single image of the virtualization software, or on a single processor, will depend on the applications that run in the virtual machines and their need for resources. It also depends on the underlying virtual machine platform, and its tolerance for over-subscription of resources. Every approach has benefits and limitations.

Conclusion

The use of virtual machines is a well-seasoned strategy that can be effectively used by any data center, large or small.

- It can support business agility via a more rapid and easily repeated mode of deployment;
- It can support operational savings (often considerable) by allowing resources to be more effectively used by multiple applications; and
- It can support business process optimization by co-locating applications to drive latency out of process.

All these goals are high on organizational to-do lists. It is time to address them with virtual machines.



[5] The tight integration of z/VM with the underlying System z hardware provides exceptional levels of isolation across all system resources.

[6] The usefulness of this approach obviously rests on securing usage-based licensing.

[7] For more on z/VM, see **the Clipper Group Navigator** entitled *Oh the Things You Can Do with z/VM 5.3*, dated February 25, 2007. It is available at http://www.clipper.com/research/TCG2007XXX.pdf.

About The Clipper Group, Inc.

The Clipper Group, Inc., is an independent consulting firm specializing in acquisition decisions and strategic advice regarding complex, enterprise-class information technologies. Our team of industry professionals averages more than 25 years of real-world experience. A team of staff consultants augments our capabilities, with significant experience across a broad spectrum of applications and environments.

- ***The Clipper Group can be reached at 781-235-0085 and found on the web at www.clipper.com.***

About the Author

***Anne MacFarland* is Director of Data Strategies and Information Solutions for The Clipper Group.** Ms. MacFarland specializes in strategic business solutions offered by enterprise systems, software, and storage vendors, in trends in enterprise systems and networks, and in explaining these trends and the underlying technologies in simple business terms. She joined The Clipper Group after a long career in library systems, business archives, consulting, research, and freelance writing. Ms. MacFarland earned a Bachelor of Arts degree from Cornell University, where she was a College Scholar, and a Masters of Library Science from Southern Connecticut State University.

- ***Reach Anne MacFarland via e-mail at Anne.MacFarland@clipper.com or at 781-235-0085 Ext. 128. (Please dial "128" when you hear the automated attendant.)***

Regarding Trademarks and Service Marks

The Clipper Group Navigator, **The Clipper Group Explorer**, **The Clipper Group Observer**, **The Clipper Group *Captain's Log***, **The Clipper Group Voyager**, **Clipper Notes**, and *"clipper.com"* are trademarks of The Clipper Group, Inc., and the clipper ship drawings, *"Navigating Information Technology Horizons"*, and *"teraproductivity"* are service marks of The Clipper Group, Inc. The Clipper Group, Inc., reserves all rights regarding its trademarks and service marks. All other trademarks, etc., belong to their respective owners.

Disclosure

Officers and/or employees of The Clipper Group may own as individuals, directly or indirectly, shares in one or more companies discussed in this bulletin. Company policy prohibits any officer or employee from holding more than one percent of the outstanding shares of any company covered by The Clipper Group. The Clipper Group, Inc., has no such equity holdings.

Regarding the Information in this Issue

The Clipper Group believes the information included in this report to be accurate. Data has been received from a variety of sources, which we believe to be reliable, including manufacturers, distributors, or users of the products discussed herein. The Clipper Group, Inc., cannot be held responsible for any consequential damages resulting from the application of information or opinions contained in this report.