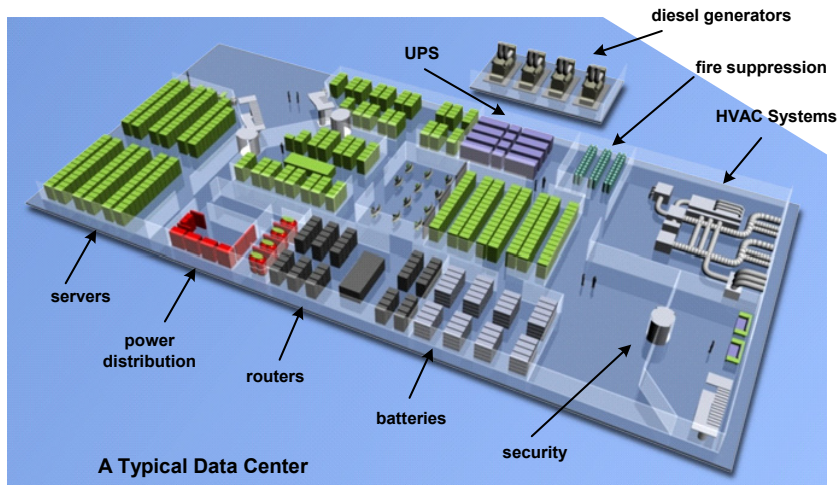


Fault Tolerance for Virtualized Environments

March 2008

Introduction

Not only do businesses today depend upon information technology (IT) for their very existences, but IT costs have become a major part of an enterprise's budget. As corporate data centers become bigger and bigger, often supporting thousands of servers, their costs for hardware, space, administration, and energy are rapidly increasing. In fact, their energy requirements sometimes bypass the available energy in their areas – remember the California brownouts of 2000?



Data Center Consolidation with Virtualization

However, a fortunate trend is evolving. Servers are becoming ever more powerful. Moore's Law states that server capacities will double every eighteen months,¹ and this trend not only has held for decades but is projected to hold well into the future. The result is that data-center servers are carrying less and less of their rated capacity. In fact, recent studies have shown that typical servers in a data-center environment that is governed by a one-application, one-server policy are running at only 10% to 15% of capacity.

If only we could harness this excess capacity, we could significantly reduce the number of servers in a data center by a factor of two, three, or even more. This would result in less hardware, less maintenance, less administration, less space, and less energy – in short, less cost by a large factor. This is the promise of virtualization.

¹ This is the common quote. Gordon Moore actually said that transistor density would double every two years.

What is Virtualization?

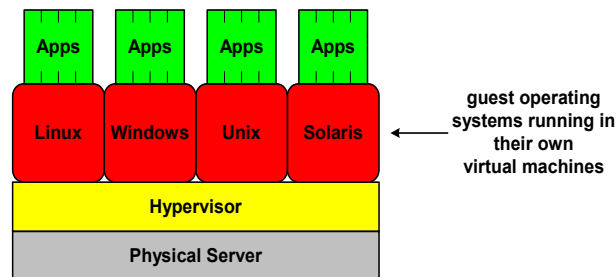
Virtualization is an architecture in which access to a single underlying piece of hardware, like a server, is coordinated so that multiple guest operating systems (virtual machines) can share that single piece of hardware with no guest operating system being aware that it is sharing anything at all.² Simply put, virtualization allows a single physical server to be partitioned into multiple *virtual machines* (VMs) that can be independently used by *guest operating systems*. As a result, the utilization of physical servers in a data center can be increased from today's average of 15% or less to 70% or more. This workload consolidation can significantly reduce the number of servers required in the data center.

Virtualization Drives the Need for Server Availability

Virtualization makes hardware availability more important than ever before. If a physical server fails, it takes down only the application that is running on it. If the application is not mission-critical to the enterprise, this may be acceptable. However, when a virtualized server fails, it takes down multiple virtual machines and all of the applications they are running. The consequences to a business can be severe, especially if some of those applications are mission-critical. The failure consequences argue strongly for physical servers that simply will not fail. This is the realm of fault-tolerant servers, which are designed to prevent hardware and software failure.

An important characteristic of a virtual machine is that it is independent. It is totally isolated from the other virtual machines just as if it were running in its own separate physical processor. Any fault in an application or guest operating system in one virtual machine is completely transparent to the other virtual machines running on that physical processor and can have no impact on them.

This implies that there must be some kind of adjudicator that controls the access by the various virtual machines to the resources of the physical server - the processor, its memory, its data storage devices, and its I/O channels. This adjudicator is known as the *hypervisor*. The hypervisor traps guest operating system calls to the processor, memory, data storage devices, and network connections and allows only one virtual machine at a time to execute these calls. In effect, it is multiplexing the access of the various virtual machines to the underlying physical processor, thereby ensuring that each gets the resources it needs.



A Virtualized Server

The Many Benefits of Virtualization

The standardization of servers on a relatively inexpensive common chip architecture – the x86 class of microprocessors, is pushing the many advantages of virtualization into the mainstream. These include:

- **Better Hardware Utilization:** As Moore's Law continues to predict, server capacity is increasing at a rapid rate. Virtualization allows a data center to use the excess capacity in today's servers to run the load of many servers as virtual machines on a single physical server.
- **Server Consolidation:** Since virtualization allows the functions currently being performed by several current servers to be consolidated onto one server, the data center requires fewer physical servers.

² Bernard Golden, *Virtualization for Dummies*, Wiley Publishing Inc., 2007.

- **Less Hardware Maintenance:** The fewer the servers, the less the maintenance workload. Fewer spare parts and fewer upgrades.
- **Reduced System Administration:** True, the administration of applications remains. However, fewer physical systems need to be administered. Typical industry experience is that data-center administration costs can be cut by 30% to 50%.
- **Reduced Space Requirements:** Large server farms can take up a lot of expensive space. By significantly reducing the size of the server farm, the space required to house the server farm is correspondingly reduced.
- **Reduced Emergency Power Needs:** Fewer servers, less power. Not to mention things like UPS (uninterruptible power supply) and other back-up and disaster requirements are proportionately less.
- **Reduced Energy Costs:** With fewer servers and reduced HVAC and lighting requirements, the amount of energy demanded by the data center is dramatically reduced. With energy prices rapidly escalating, this can be a significant operational cost savings. More important to some companies is the positive environmental impact that reducing energy consumption can have. Virtualization is green!
- **Reduced Capital Expenditures:** Less money needs to be invested in server hardware, data-center space, and HVAC and lighting infrastructure.
- **Reduced Operating Expenses:** Less hardware maintenance and system administration, along with the significant savings in energy costs, result in dramatically reduced operating costs and a reduction in the total cost of ownership (TCO) for the data center.

With all of these benefits, why hasn't everyone adopted virtualization? One is the hesitancy to learn a new technology. The other is convincing the manager who has been running his precious application for years on his own server to now move it to a server shared by other (perhaps less well-behaved) applications.

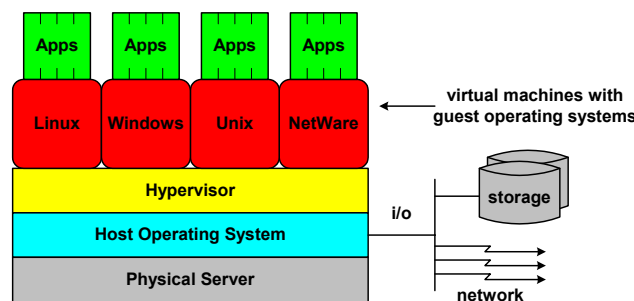
Virtualization Architectures

Let's examine the two primary ways in which virtualization is implemented in currently available products: operating system virtualization and bare-metal virtualization.

Operating System Virtualization

Operating system virtualization emulates the underlying host operating system for its guest operating systems. There are two ways that products create virtual environments on top of a host operating system:

Virtualization of the Physical Server: With this technique for virtualization, a host operating system is installed on the physical server. A virtualization layer, the hypervisor, is then installed on top of the operating system. On top of the hypervisor are created several virtual machines. Each VM can run a guest operating system. The guest operating systems may be a mix of any operating systems supported by the product, and each runs its own applications.



Virtualization with a Host OS

The hypervisor traps calls from the guest operating systems to the physical server and multiplexes access to the physical layer through the host operating system so that each virtual machine thinks that it has sole access to the physical server. The I/O devices that a virtual machine can access are those supported by the host operating system.

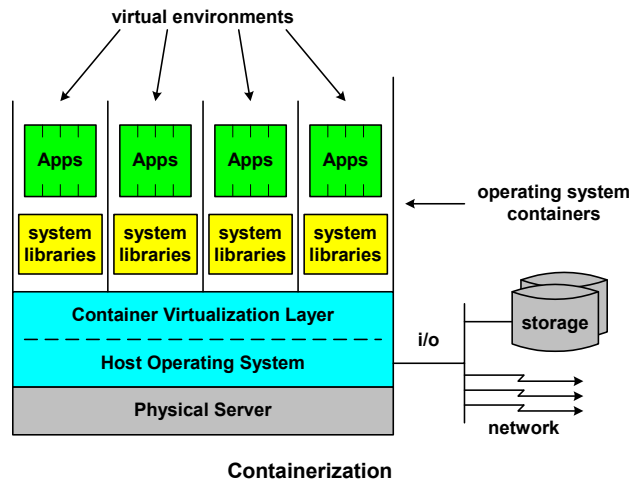
When a virtual machine is created, it appears as an empty physical server with a blank disk. The disk must first be formatted, and then a guest operating system must be installed. Thereafter, applications may be installed as if this were a stand-alone server running the guest operating system.

Therefore, once the virtualized system is configured and virtual machines are created, the administrative procedures are identical to those of separate physical servers. No special knowledge of virtualization is needed of the application administrators. This includes the installation of the operating system, the installation and management of the applications, and the management of data storage, whether used for files or for a relational database.

One concern with this method of virtualization is that if the host operating system should crash, all virtual machines running on the physical server are taken down. Since operating system crashes tend to be more frequent than physical server crashes, this creates an availability issue.

Virtualization of the Host Operating System: This technique is also known as containerization. As opposed to creating virtual views of the underlying physical server, containerization provides virtual views of the underlying host operating system. Rather than creating virtual machines, containerization creates independent containers that provide a virtual environment reflecting the underlying operating system.

The host operating system is typically some version of Windows® or Linux®. A virtualization layer within the operating system creates containers, each containing a set of system libraries called by the applications in each container. The system libraries replace the guest operating system normally installed in a virtual machine.



Application calls to the operating system are intercepted by the system libraries in the container occupied by the application. These libraries emulate the underlying operating system. So far as the applications are concerned, they are interfacing directly with the host operating system as if they had exclusive use of it.

One limitation of containerization is that all applications see the same host operating system. Therefore, they must all be configured to run under that operating system. Their configurations must match the host operating system's version and even its patch level. Therefore, pure virtualization is not provided since there is not the freedom to run any version of a guest operating system and its applications on the physical server.

As with operating system virtualization that exposes the physical server, as described above, a crash of the operating system in this case will take down the entire virtualized environment.

An example of a containerization product is Virtuozzo™ from SWsoft® (www.swsoft.com). It supports both Windows® and Linux® host operating systems. Though the Windows® version is reflected in the virtual environment so that all applications must run under that version of Windows®, the use of Linux® is more flexible in that the applications are not so severely restrained and may run some different versions of Linux®. Virtuozzo™ is also available as open source (OpenVZ).

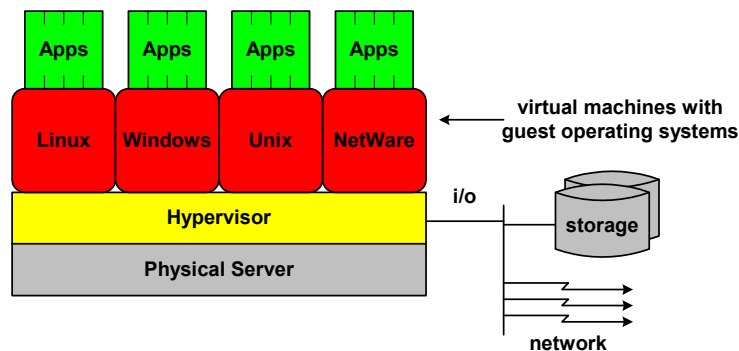
In addition, Sun's Solaris operating system supports containerization (see www.sun.com/datacenter/consolidation/virtualization).

Bare-Metal Virtualization

Bare-metal virtualization emulates the underlying physical server for its guest operating systems. Bare-metal virtualization does not use an underlying host operating system as with operating system virtualization. Rather, the hypervisor sits directly on top of the physical server (thus the name bare metal). There are two forms of bare-metal virtualization in use: bare-metal hypervisor, and paravirtualization.

Bare-Metal Hypervisor: A bare-metal hypervisor runs directly on the physical server. It establishes virtual machines by providing an emulated hardware interface to each virtual machine, each of which can be running a different guest operating system. It traps calls made by the guest operating systems and multiplexes them to the physical processor. It handles all device interrupts and device management.

The bare-metal hypervisor can support dissimilar guest operating systems running in its virtual machines. Typically, different versions of Windows®, Linux®, and UNIX®, among others, can be running simultaneously in the different virtual machines so long as they are designed to run in an x86 architecture. However, the guest operating systems can only use those device drivers supported by the hypervisor.



Bare-Metal Virtualization

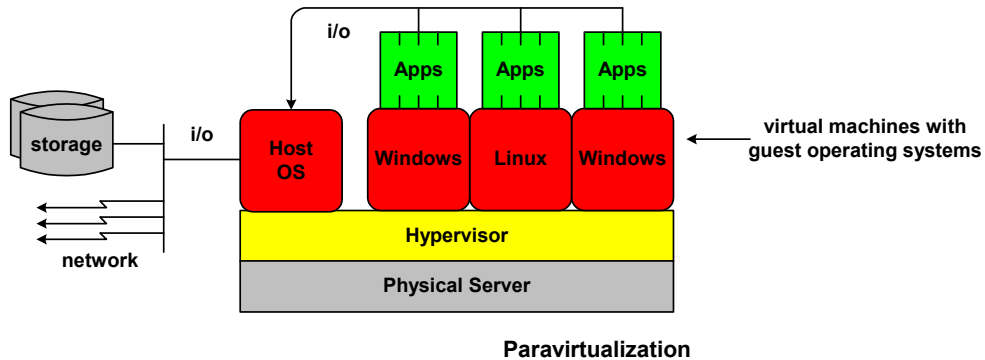
A technical challenge with bare-metal hypervisors is that the calls to server hardware made by guest operating systems must be modified to calls to the hypervisor instead. Rather than requiring special virtualized versions of the operating systems, the hypervisor at run time modifies the calls in the standard guest operating system to call the hypervisor instead. This process is called *runtime binary translation*. A limitation of this approach is that only those guest operating systems that are supported by the hypervisor's runtime binary translation can be used in the virtual machines.

The leading example of a bare-metal hypervisor is ESX™ Server from VMware®. ESX™ supports Windows®, Linux®, UNIX®, and Novell NetWare® guest operating systems. It is available as an embedded component built into the server hardware.

(See http://www.vmware.com/files/pdf/esx_datasheet.pdf).

Paravirtualization: This technology uses a hypervisor similar to the bare-metal hypervisor described above except that the paravirtualization hypervisor only provides emulation of the underlying physical server's processor and memory. I/O devices are supported via a privileged guest operating system that is running in its own virtual machine.

The device drivers in other guest operating systems are replaced with stubs that communicate via shared memory with stubs in the privileged guest operating system for device access. In effect, the privileged guest operating system acts as a gateway to its devices for other guest operating systems. Any device supported by the privileged guest operating system can be used by guest operating systems running in the other virtual machines.



Paravirtualization

Paravirtualization brings with it two advantages over the bare-metal hypervisor previously described:

- A wider range of I/O devices can be used since special device drivers do not have to be developed for the hypervisor. Any device supported by the privileged guest operating system can be used.
- Much lower overhead is required to support the virtual machines since the hypervisor does not have to support I/O.

A leading example of paravirtualization is Xen[®] (originally XenSoft[™]) from Citrix[®] (www.citrixserver.com). Xen[®] runs on a wide variety of hardware platforms and imposes a very small footprint. The entire Xen[®] code base is less than 50,000 lines of code. Xen[®] supports Linux[®] and Windows[®] guest operating systems. Xen[®] is also available as open source (www.xen.org).

Availability Issues with Virtualization

Typical virtualization products come with some availability features. They are based on the monitoring of the operational state of the virtual machines by the hypervisor.

Virtual Machine Failover

At the basic level is failover. Should a virtual machine fail, the hypervisor will detect that and will restart the virtual machine. All work-in-progress is lost, but the application is restored in minutes or less.

This level of failover protects against a virtual machine or a guest operating system crash, but it does not protect against a crash of the underlying physical server. Should the server crash, all virtual machines that were running on the server are, of course, lost.

Server crashes can be handled by pairing virtualized servers in much the same manner as clusters. This solution requires that the application databases be on network attached storage (NAS) or a storage area network (SAN) so that they can be generally accessible from multiple physical servers. Failover is accomplished via inter-hypervisor coordination on the various physical servers involved.

Thus, if a physical server fails, its virtual machines can be migrated to other physical servers. It is not necessary that these other servers be idle standbys. They may be managing their own active virtual machines. The only requirement is that they have capacity available to pick up some or all of the load of a failed server.

When a physical server fails, its virtual machines are migrated by the hypervisors to other physical servers and are distributed among these servers to balance the new load profile. The servers to which the failed virtual machines have migrated have access to the networked application databases, and the migrated applications can continue to function. As with clusters, failover can take several minutes; and all work-in-progress is lost.

Another option is *server pooling*. In this configuration, several virtualized physical servers are organized in a pool that itself is virtualized. To outside users, the server pool appears as a single virtualized server.

A specific virtual machine can be resident on any of the physical servers. Moreover, it can be moved from server to server under control of the pooling management facility without user interruption. This is useful for load balancing. If the load on one physical server should climb to an uncomfortable level, the pooling manager can automatically move it to another server. During this process, application state is maintained so that no work-in-progress is lost due to the move.

Pooling configurations bring another availability benefit, and that is eliminating planned downtime for software or hardware upgrades. If the hypervisor is to be upgraded, the virtual machines are moved to another server, the upgrade is performed, and the virtual machines are then moved back, a process that is transparent to the users. If a guest operating system is to be upgraded, a new virtual machine is created, the upgraded operating system is installed, and the applications are moved from their old virtual machine and guest operating system to the new configuration, all without user interruption or lost work.

Server pooling is the first step in utility computing, wherein applications are run by reservation when and only when they are needed. Products that support the transparent movement of VMs from one physical server to another include VMotion from VMware® and XenMotion® from Citrix®.

In summary, virtualization failover can eliminate planned downtime because application state can be maintained as virtual machines are moved from one operating environment to another. However, virtualized failover cannot prevent unplanned downtime. Though the failed virtual machines can be restarted, work-in-progress is lost; and it can take several minutes or more to return the failed virtual machines to service.

This is high availability, not continuous availability.

The Requirement for Fault Tolerance

In a one-application, one-server environment, if a server fails, that application fails. The pain is felt, but it is limited. However, if a virtualized server (that is, a server supporting several virtual machines) fails, all of the applications running in the virtual machines on that server are down. This is a pain of a greater magnitude. If the server is running many mission-critical applications, the pain could well be intolerable.

In the general case, good practices demand that no more than one application running on a virtualized server be mission-critical. In this way, if a server fails, only one critical application is lost. The loss of the other applications for a short while is presumably tolerable.

However, this type of configuration cannot always be accomplished or be guaranteed. The mix of applications in a data center may involve so many critical applications that more than one will have to be assigned to the same physical server. Moreover, failover actions may consolidate multiple critical applications on a single server. Even worse, in a pooled environment used for load balancing, there may be no control of where an application runs. It is quite likely that at times multiple critical applications will be resident on a single server. A server crash can then take down several critical applications all at once.

This dilemma is solved by the use of fault-tolerant servers. A fault-tolerant server is one that is designed to survive any single fault and many cases of multiple faults without any service interruption or loss of work in process. A failure is completely transparent to the user.

Fault-tolerant servers have been measured in the field to have average times between failures that are orders of magnitude – up to a hundred times or more – longer than those experienced by standard high-availability servers. High-availability servers in common use tend to have availabilities of three 9s – that is, they will be up 99.9% of the time and will be down about eight hours per year. On the other hand, fault-tolerant servers experience availabilities of more than five 9s. They will be up more than 99.999% of the time and will experience less than five minutes per year of downtime.

Standard industry servers can provide high availability. Fault-tolerant systems provide continuous availability. The use of fault-tolerant servers in virtualized environments can significantly reduce the pain of server crashes taking down mission-critical applications or even groups of important but not critical applications. This is the case of the fault-tolerant server line from Stratus Technologies.

Fault-Tolerant Virtualized Servers from Stratus Technologies

Stratus Technologies of Maynard, Massachusetts (www.stratus.com) solves this problem with its line of fault-tolerant ftServer® systems. These servers provide uptimes that are orders of magnitude greater than those provided by industry standard servers. They reduce annual downtime to an average of less than five minutes per server as compared with hours of annual downtime for industry-standard servers. As opposed to the high availability that standard servers provide, fault-tolerant servers provide near-continuous availability.

With their support for Microsoft® Windows® and Red Hat® Enterprise Linux® virtual machines, ftServer systems provide a simple plug-and-play integration into existing virtualized server farms. System administration is no more difficult than that for a standard server. Furthermore, these fault-tolerant servers are priced competitively with other high-availability offerings such as clusters, which are much more difficult to set up and administer and which do not provide continuous availability.

With Stratus® ftServer systems, a data center can consolidate the critical workload of several physical servers into a much more reliable fault-tolerant platform with little effort and with a potentially lower total cost of ownership when the cost of downtime is considered.

Summary

Virtualization is an extremely important and effective technology to reduce the IT costs of data centers. It has the potential to increase server utilization from 15% or less to 70% or more. As a result, the size of the server farm can be significantly reduced, less space with its environmental HVAC controls is required, and energy usage can be cut by a large factor.

But virtualization brings with it a major problem. As opposed to the one-application, one-server model, should a virtualized server fail, many applications are brought down. If some of these are mission-critical to the organization, the cost of this downtime could be very high in terms of lost business, customer dissatisfaction, regulatory penalties, and so on.

Fault-tolerant servers from Stratus Technologies solve this problem. Through their dual modular redundancy, these servers can continue to service their users with no loss of work following any single fault and many cases of multiple faults. Their installation is plug-and-play, and they require no more administrative knowledge or effort than is required by an industry standard server. ftServer costs are comparable to other high-availability solutions such as clusters, but without the administrative difficulties and without incurring any downtime following a failure.

Especially when the cost of downtime is considered, virtualized fault tolerance can bring continuous availability to a data center at a competitive cost and with no special administrative skills.

Specifications and descriptions are summary in nature and subject to change without notice.

Stratus and ftServer are registered trademarks of Stratus Technologies Bermuda Ltd. Microsoft and Windows are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. The registered trademark Linux is used pursuant to a sublicense from the Linux Mark Institute, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis. Intel is registered trademark of the Intel Corporation in the United States and other countries. Red Hat, the Red Hat Shadowman logo and Enterprise Linux are registered trademarks of Red Hat, Inc. in the United States and other countries. VMware and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Virtuozzo, SWsoft, OpenVZ, Sun, Solaris, ESX, Novell, Netware, Citrix, Xen, XenSoft and XenMotion are either trademarks or registered trademarks and are the property of their respective holders. All other trademarks and registered trademarks are the property of their respective holders.