the Availability Digest

Fault Tolerance for Virtual Environments – Part 3

June 2008

Virtualization significantly increases the utilization of a server by creating several independent *virtual machines* (VMs) on a single physical server. To its copy of the operating system (a *guest* operating system), each virtual machine appears as if it were a dedicated physical server.

In Parts 1 and 2 of this series,[1] we described the reasons for the burgeoning interest in virtualization and how virtualization is implemented. Though virtualization can significantly reduce data-center costs, the loss of a virtualized server can mean the loss of many applications. Here in Part 3, we address the very important problem of achieving continuous availability in virtualized environments.

Virtualization products provide a broad range of failover capabilities, but all can result in long failover times as applications are migrated and as corrupt databases are repaired. This problem can be alleviated by using fault-tolerant servers to host virtualized environments. Fault-tolerant servers can withstand any single fault and many multiple faults with no impact on the user, and they can reduce the incidence of costly failovers by one or two orders of magnitude.

We conclude with some brief reviews of virtualization products and fault-tolerant servers that provide the features needed to achieve the high availability required in virtual environments.

But first, we briefly review Parts 1 and 2.

Virtualization Review

The Drivers for Virtualization

Over the years, data centers have grown to be massive. Some data centers host thousands of servers. Historically, each application often was hosted by its own dedicated server. This was necessary not only from a capacity viewpoint but by a desire of management to have control over its own IT resources.

However, Moore's law has prevailed over the years. In 1965, Gordon Moore, one of the founders of Intel, stated: "The complexity for minimum component costs has increased at a rate of roughly a factor of two per year."[2] In effect, he was saying that the density of transistors on a chip would double every two years (this is now often quoted more conservatively as a doubling every eighteen months). Moore's law not only continues to be valid today, more than four decades later, it also applies across the board to processor power and storage capacity alike.

[1] Fault Tolerance for Virtual Environments – Part 1, *Availability Digest*; March 2008.
Fault Tolerance for Virtual Environments – Part 2, *Availability Digest*; April 2008.
[2] Gordon E. Moore, Cramming more components onto integrated circuits, *Electronics*, Volume 38, Number 8; April, 1965.

Consequently, data-center servers today are much more powerful than their forbearers; but they continue to run the same old applications. As a result, the average utilization of servers in a data center is often in the 10% to 15% range.

Virtualization allows the consolidation of these servers, with each server running as a virtual machine within a common physical server. Thus, data-center physical-server utilization can be increased from 15% or less to 70% or more, reducing the physical server count by a significant factor.

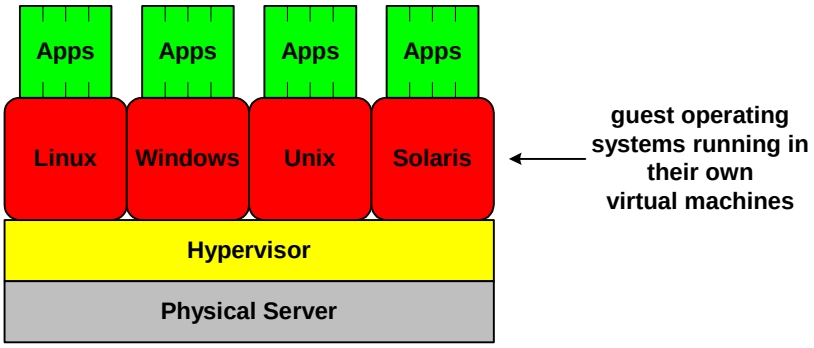
Fewer servers mean less capital cost, less maintenance, less administration, less space, less cooling and lighting, lower UPS requirements, and less energy consumed. In short, the capital costs and operating costs for a data center can be significantly reduced, often by a factor of four or greater.

How is Virtualization Implemented?

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.[3] Simply put, virtualization allows a single physical server to be partitioned into multiple *virtual machines* (VMs) that can independently be used by *guest operating systems*.

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

Why is this technology just now becoming widely available? It is because of the standardization of industry-standard servers on a relatively inexpensive common chip architecture – the x86 class of microprocessors. This has allowed the development of hypervisors to virtualize a common hardware architecture rather than having to support multiple such architectures.

[3] Bernard Golden, *Virtualization for Dummies*, Wiley Publishing Inc.; 2007.

There are two ways in which virtualization is implemented with today's products:

- Operating System Virtualization, in which the virtualization layer sits on top of a *host operating system* that is installed on the physical server. The host operating system provides the interfaces between the virtual environments and the physical processor and its I/O devices.

- Bare-Metal Virtualization, in which the virtualization layer (the hypervisor) sits directly on top of the hardware with no intervening host operating system. The hypervisor in this case provides the common device drivers.

These architectures are described in some detail in Part 2 of this series, and current products providing these features are noted.

Virtualization and Availability

But virtualization comes with a price, and that price is availability. If a classic 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, if a virtualized server fails, it takes down the equivalent of several servers since each virtual machine hosted on the virtualized physical server fails. Thus, the failure of a virtualized physical server will take down many applications and is far more painful to the enterprise, especially if some of these applications are mission-critical.

Therefore, it is imperative that there be some sort of failover mechanism so that virtual machines can continue to function in the event of the failure of a physical server. Furthermore, the failure consequences of a virtualized physical server argue strongly for physical servers that simply will not fail – at least not very often. This is the realm of fault-tolerant servers, which can survive any single fault as well as many cases of multiple faults.

We now look at the various mechanisms available in virtualization products to ensure availability.

Virtual Machine Failover

Typical virtualization products come with some availability features. Almost all hypervisors monitor the operational state of the virtual machines running on their physical servers.

At the basic level is failover. Should a virtual machine fail, the hypervisor will detect that and will restart the virtual machine on the same physical server. Failover can often be accomplished in seconds because the image of the entire virtual machine can be stored in a file and restored by the hypervisor very quickly.

All work-in-progress is lost, a common result of server crashes. This usually means that requests in progress must be resubmitted to the restored server.

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.

Clustering

Physical server crashes can be handled by pairing virtualized servers in much the same manner as contemporary clusters. This solution requires that the application databases be on network attached storage (NAS) or on a storage area network (SAN) so that they can be generally

accessible from multiple physical servers. Failover is accomplished via interhypervisor 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.

A special layer of coordinating software monitors all of the hypervisors and their virtual machines. If it sees that a hypervisor on one physical server is not responding, it restarts any virtual machines that were running on the failed hardware on one or more other physical servers. The restarted virtual machines may be distributed among surviving 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 to hours as applications are started and as corrupted databases are repaired. All work-in-progress is lost.

In addition, operating VMs can be migrated to other physical servers without interruption to support load balancing and to allow maintenance and upgrades on a physical server with no user downtime.

Server Pooling

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.

In summary, virtualized pooling 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 due to a physical server failure. Though the failed virtual machines can be restarted on surviving servers, 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 over 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.

Disaster Recovery

Disaster recovery is the capability to continue operations even if an entire data center is lost. Today, there are no disaster recovery products specific to virtualization. Rather, standard disaster recovery techniques can be employed. The first step, of course, is to have two data centers that are geographically separated. Each may have its own set of virtualized servers so long as there is enough spare capacity to handle the load of the other data center should that data center fail.

Some fault-tolerant systems allow the redundant processors to be separated, in some cases up to a few miles. If greater separation is needed to prevent a common disaster from taking out both data centers (as is usually the case), asynchronous replication engines may be used to maintain a reasonably current copy of the application databases at each site so that applications may be restarted on servers at the surviving site in the event of a disaster.

Fault-tolerant systems generally provide some sort of disaster-recovery facilities. Note that disaster recovery is different from disaster tolerance. Disaster recovery means recovering from a disaster. Disaster tolerance means to be unaffected by a disaster. Active/active systems[4] provide

[4] What is Active/Active?, *Availability Digest*; October, 2006.

disaster tolerance, as recovery can be accomplished so fast that users are unaware of the disaster.

Virtualization Products for High Availability

All of the functions required for high availability described above are provided by one or more current products.

Virtual Machine Failover

Virtually all virtualization products today support virtual machine failover. Should a VM fail in an otherwise operable physical server, the hypervisor will detect that and will restart the VM in that server.

Such products include:

- VMware ESX server
- Xen (open source) and XenSource (from Citrix)
- Virtual Iron (based on the Xen hypervisor)
- Sun Solaris operating system
- SWsoft from Virtuozzo

Clustering

Clustering provides failover of virtual machines from a crashed physical server to one or more surviving physical servers that may be running their own VMs. VMs may also be failed over to support load balancing and maintenance and upgrade activities with no interruption to the users. These products include:

- VMware VMotion
- Virtual Iron LiveMigration

Server Pooling

Server pooling organizes several virtualized physical servers into a pool that itself is virtualized. To outside users, the server pool appears as a single virtualized server. VMs can be freely (and automatically) moved around the pool for load balancing, and they can be moved to eliminate downtime during planned maintenance and upgrades. VMs that were resident in a failed physical server can be moved to one or more surviving physical servers.

Products that provide server pooling include:

- VMware DRS (Distributed Resource Scheduler)
- Virtual Iron LiveMigration

Fault-Tolerant Servers

Even with clustering and pooling, the failure of a physical server can mean minutes to hours of downtime for its hosted applications as virtual machines are moved to surviving servers, applications are restarted, and corrupt databases are repaired. The use of fault-tolerant servers, which can survive the failure of any single component as well as many multiple failures, can reduce the frequency of such outages by an order of magnitude or more.

HP NonStop servers (formerly Tandem Computers) are the granddaddy of fault-tolerant virtualization. Since the late 1970s, these systems have been presenting a single-system image of multiple applications running over hundreds of processors.

However, in this series, we are interested in the virtualization of industry-standard servers and Windows, Unix, and Linux applications. To be useful for virtualization, fault-tolerant servers must be x86-based since all current hypervisors today expect an x86 platform. Products that can be used as virtualized fault-tolerant platforms include the following.

Stratus ftServer

Stratus Technologies of Maynard, Massachusetts, (www.stratus.com) provides the fault-tolerant ftServer.[5] The ftServer comprises two processors with redundant I/O channels running in lock-step. Included in the ftServer is 1.5 terabytes of mirrored (RAID 1) disks. Therefore, network attached storage (NAS) or storage area networks (SAN) are often not necessary. The direct-attached storage of the ftServer is fully redundant and can withstand any single failure. Stratus posts operational availabilities on their home page as measured by current field failures. More than five 9s of availability are typically displayed.

Stratus has integrated VMware's ESX server into their line of fault-tolerant ftServers. As a bare-metal hypervisor, ESX runs directly on top of the ftServer processor. Guest operating systems supported by ftServer include Windows and RedHat Linux.

For disaster tolerance, Stratus uses the Double-Take asynchronous replication engine to replicate Windows databases to a remote site, and it uses the GoldenGate replication engine to replicate Linux databases.

Stratus Avance

Stratus' newly-announced Avance fault-tolerant product uses a pair of industry-standard servers in a fault-tolerant configuration. The servers can be running either Windows or Linux. As transactions are executed on one system, changes to the database are replicated synchronously to the other system over a one gigabit/second private link. The two systems can be separated by up to a half a kilometer. Via Stratus' professional services, a third node can be provided as a disaster recovery site at any distance using asynchronous replication.

Should its database mirror fail, the active processor can use the mirror on the backup system over the high-speed intersystem link. Should the active processor fail, users are routed to the backup system, which now becomes the active system. Because of the synchronous data replication, no data is lost following the failure of the active system.

Stratus has integrated the Citrix Xen hypervisor into Avance to provide the support for a virtualized environment. With Avance, failover of the entire virtual environment is simply a matter of rerouting users.

Marathon everRun

everRun FT, from Marathon Technologies (www.marathontechnologies.com of Littleton, Mass., uses two standard Windows servers integrated with Citrix's XenSource hypervisor to provide fault-tolerant virtualization. The servers are interconnected by a dual gigabit Ethernet channel. The servers execute the same code at the same time and are synchronized at the instruction level. This means that each server will write the same data to its disk subsystem at the same time as the other server. Therefore, database replication is synchronous; and no data is lost should one server fail.

[5] Fault-Tolerant Windows and Linux from Stratus, *Availability Digest*; September 2007.

In the event of a server failure, the surviving server continues in operation with no impact to the user. When the failed server is returned to service, its database is resynchronized with the active database; and fault-tolerant operation continues.

The two servers can be separated by several miles for disaster tolerance.

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.

Many virtualization products bring failover capabilities to a virtualized data center. However, failover following a physical server crash can take minutes to hours as applications are brought up and as corrupted databases are repaired.

Fault-tolerant servers 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. 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.