the Availability Digest

www.availabilitydigest.com

Leveraging Virtualization for Availability

December 2010

Virtualized environments are becoming commonplace in today's data centers. Since many virtual servers can be hosted on a single physical server, the server count in the data center along with the associated savings in floor space, energy, and administration can reduce the server component cost in a data center by 80% or more.

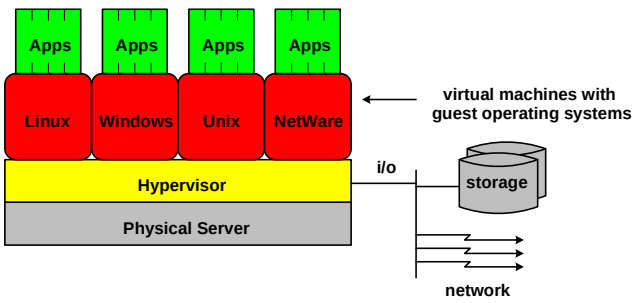
However, if a server should fail, not one but many applications are taken down. Availability is a critical issue in a virtualized environment. Fortunately, there are several ways to provide high availability in these environments. In principle, even continuous availability can be achieved.

What is Virtualization?

Until recently, it has been the practice in many data centers to have a separate server for each application. After all, a business-unit manager did not want to share his server with other applications that he could not control. Who knows what bugs lay in the applications of others or what horrible performance hits they might impose?

However, as these servers were upgraded to more powerful servers over the years, they became less and less loaded. In many data centers today, it is not uncommon to find that the Windows, Linux, and UNIX servers in the data center are running at only 10% to 20% of capacity. What a waste of computing power.

Virtualization allows the consolidation of multiple physical servers onto one physical server - the *host* - as *virtual machines (VMs*). Each virtual machine appears to the outside world as if it were an independent physical server. Each is independent of the others – a fault in one will not affect the others (though the total of the loads imposed by the set of VMs must be within the capacity of the host). Each VM provides its own operating environment, running its own copy of a *guest operating system*. Typical virtualization products support Windows, Linux, and UNIX operating systems.



Virtualization

A *hypervisor* adjudicates requests from the various VMs for host services, such as input/output devices.

Leading virtualization products include ESX from VMware, Xen from Citrix, and Hyper-V from Microsoft.

The Virtual Availability Problem

A major problem with virtualization is availability. Before virtualization, if a server failed, one application was lost. In a virtual environment, if a physical host fails, it takes down multiple applications with it. There are some strategies that can be used to mitigate the impact of a host failure.

One strategy is to allocate critical applications to different hosts so that if a host fails, only one critical application is affected. However, if load balancing is used, this strategy becomes ineffective. With load balancing, should one host become heavily loaded, one or more of its virtual machines are moved to other hosts to rebalance the load. Therefore, over a period of time, the allocation of applications to hosts will become indeterminate. There is nothing to prevent two or more critical applications from being hosted on one physical host. Should that host fail, multiple critical applications might fail as well.

Another strategy is to use fault-tolerant servers as hosts. This will typically reduce the failure rate of a host by a factor of ten or more. Typical industry-standard servers today have an availability of three 9s, which translates into one or two failures per year. Stratus' Avance[1] and Marathon's everRun both support the Xen hypervisor and have availabilities of four 9s, which translates into failure intervals of five to ten years. Stratus' ftServer[2] has a field experience of more than five 9s (a failure interval in the order of decades). ftServer supports VMware.

Virtual Availability Solutions

Beyond these strategies, the virtualization products themselves provide several features for improving availability in the face of several failure modes.

The first requirement for improving availability is to know when a VM has failed (of course, if a host fails, all of its VMs will fail). To accomplish this, each VM generates a heartbeat. A Virtualization Manager on the host monitors the heartbeats of its VMs.

Virtual Machine Failure

If a VM's heartbeat is lost and the VM has had no disk or network activity, the VM is restarted. All work in progress is lost.

What does it mean to restart a VM? A VM is represented physically as a file on the host's storage device, much as the operating environment on a PC is stored. To restart the VM, all that is required is to reboot the VM from its disk image as stored on the host. The hypervisor ensures that the new VM is totally independent of other VMs that are currently running.

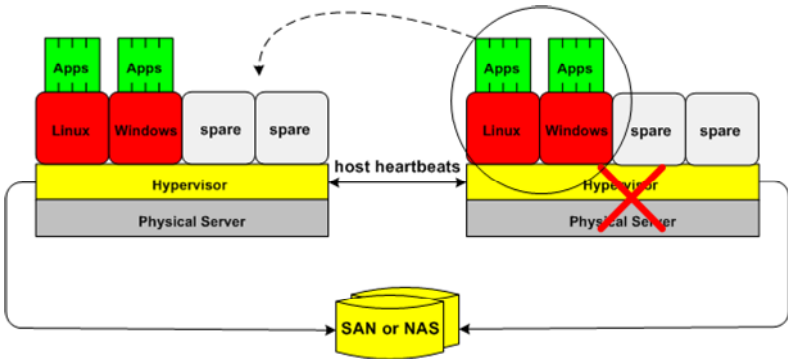
---

[1] Stratus Avance Brings Availability to the Edge, *Availability Digest*; February 2009.
http://www.availabilitydigest.com/public_articles/0402/avance.pdf

[2] Fault-Tolerant Windows and Linux from Stratus, *Availability Digest*; September 2007.
http://www.availabilitydigest.com/public_articles/0209/stratus.pdf
Stratus Bets $50,000 That You Won't Be Down, *Availability Digest*; January 2010.
http://www.availabilitydigest.com/public_articles/0501/stratus_guarantee.pdf

Host Failure – High Availability

In order to recover from a host failure, there must be two or more hosts that are clustered together. Clustering in this case simply means that all hosts share the same storage device – typically a SAN (storage area network) or a NAS (network-attached storage). Keep in mind that a VM is simply a file on the SAN or NAS storage. Furthermore, the hosts must be connected with a redundant high-speed data link.



Host Recovery
Rolling Upgrades

Just like the VMs, the hosts generate heartbeat messages that are monitored by the Virtualization Managers on each host. Should a host fail, the failed VMs are restarted on a surviving host. The only criterion is that the host to which the failed VMs are moved has to have enough capacity to handle the increase in VM load. If several hosts are clustered, the failed VMs can be redistributed across multiple surviving hosts. Again, all work in progress is lost.

An example of a facility providing host failover is VMware HA from VMware.

Eliminating Planned Downtime

A cluster can also be used to eliminate planned downtime by rolling upgrades through the cluster.

The first step is to move the VMs from the host to be upgraded to other hosts. The idled host can then be taken down and upgraded. Its VMs are moved back following the upgrade. This process is repeated for other nodes that must also be upgraded.

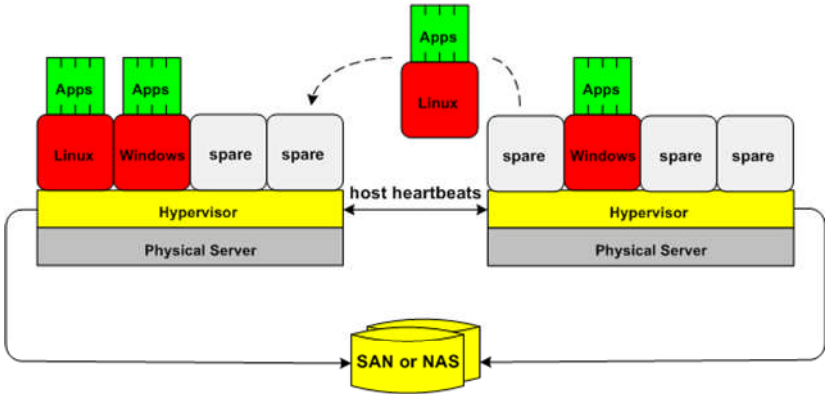
This controlled move is transparent to the users. No work is lost. VMotion from VMware is an example of a facility providing this capability.

Unlike classic clusters, there is no requirement for the nodes in a virtualized cluster to be identical. They can be different hardware platforms so long as they use a common hypervisor and have sufficient capacity to fulfill their backup roles.

Load Balancing

Because VMs can be easily moved from one host to another in a cluster, a complex of virtual machines and hosts can be easily load balanced. All that is required is to move one or more VMs from a heavily loaded host to other hosts in the cluster. As with rolling upgrades, this is accomplished transparently to the users. No work is lost.

Most virtual environments provide a load balancer that monitors the loads on the hosts in the cluster. It makes the load-rebalancing decisions. An example of a load balancer is VMware's Distributed Resource Scheduler.



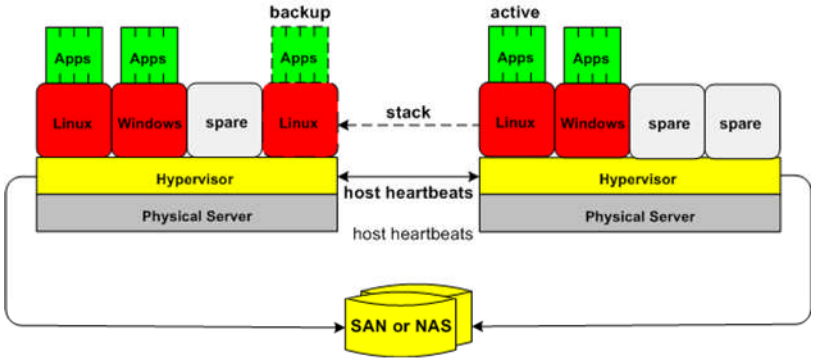
Load Balancing

Host Failure – Fault Tolerance

The problem with the availability approaches discussed so far is that failover can take minutes or more, and all work in progress is lost. This can be mitigated with a fault-tolerant approach to VM recovery available in some virtualization products.

The strategy is to provide a hot backup of a VM on another host. Should a VM fail, its backup automatically and seamlessly takes over. Failover is measured in seconds, and no work is lost.

VMware Fault Tolerance is one example of such a product. It creates a backup copy of a VM to be protected on another host. During operation, the stack of the active VM is replicated over a high-speed local channel to the backup VM. Therefore, the backup VM is always in the same state as the active VM. If a host fails, the backup VM immediately takes over processing, transparently to its users.



Host Fault Tolerance

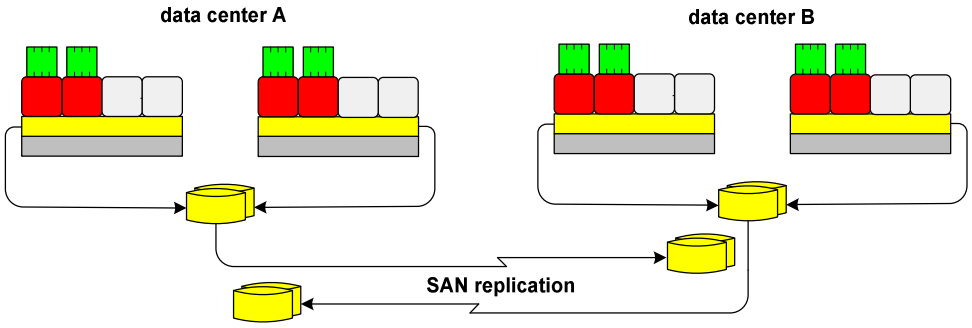
Note that this facility does not protect against a VM failure on an otherwise operable host. This is because the VM has somehow become corrupted, and the stack of its backup will be similarly corrupted. In this case, the VM must still be rebooted.

Disaster Recovery

None of the above architectures provides disaster recovery. The clustered hosts and common storage subsystem must all be collocated in the same data center. To provide recovery from a

disastrous event that disables the data center, a disaster-recovery site needs to be provided. This is typically another site that has its own set of hosts and storage subsystems.

The physical servers in both data centers can be active hosting their own sets of virtual machines. Each replicates its database to a copy at the other data center (SAN or NAS replication is typically unidirectional). Should a data center be taken out of service, the cluster in the surviving data center brings the replicated database into a consistent state; and the physical servers then mount it. At this point, the downed VMs can be restarted on hosts in the surviving cluster.



Disaster Recovery

As with typical disaster-recovery sites, recovery can take hours; and all work in progress at the failed data center is lost. An example of a disaster-recovery facility is VMware's Site Recovery Manager.

Continuous Availability

Continuous availability has not yet made it into the virtualized world in terms of existing products. Recovery from failures takes many minutes to hours.

However, conceptually, there is no technical reason why continuous availability could not be achieved in a virtual environment. What is needed is bidirectional replication between the data stores at the various sites. In this way, all hosts can be actively processing transactions for the same application. If a VM, a physical host, or an entire data center should become disabled, all that needs to be done is to resubmit failed transactions to surviving VMs. Recovery can be in seconds.

Summary

In the early days of virtualization, availability was a serious concern because of the "all your eggs in one basket" syndrome. However, as the technology has matured, several facilities have been developed by the virtualization vendors to provide a wide range of high-availability options.

Though we have used the VMware options[3] as examples, most of the major virtualization products provide equivalent functionality. Citrix's Xen 4.0 now includes fault tolerance along with the standard high availability options.[4] Microsoft's Hyper-V provides Quick Migration and Live Migration for recovering or moving VMs and Stretch Clustering for disaster tolerance.[5] Microsoft and Citrix also play together via Citrix's Essentials for Microsoft Hyper-V, which adds significant management functionality to Hyper-V.[6]

---

[3] Mike Laverick, VMware High Availability, *VMware/Intel Presentation*.
[4] http://xen.org.
[5] High Availability and Disaster Recovery Considerations for Microsoft Hyper-V, Microsoft Tech-Ed; 2009.
[6] Citrix Essentials for Hyper-V – Express Edition, *The Citrix Blogs*. http://www.citrix.com/English/ps2/products/subfeature.asp?contentID=1855667.