Global Knowledge®

Expert Reference Series of White Papers

PowerVM Virtualization Essentials

PowerVM Virtualization Essentials

Iain Campbell, UNIX/Linux Open Systems Architect, eLearning Specialist

Introduction

Today, all major processing hardware platforms support the ability to create virtualized instances of a single server. IBM's proprietary POWER (Performance Optimized With Enhanced RISC) architecture is no exception; the complete virtualization package encompassing all necessary components is termed PowerVM.

While the basic concept of the virtual machine is generic, each specific implementation has its own architecture and associated terminology. In this paper we will present an overview of the PowerVM architecture indicating the relative place and function of each of the major components.

We will start with a big-picture look at the architecture and then introduce some of the functionality offered by this platform.

The Big Picture

The major architectural components and terms are illustrated in Figure 1. The key components are the Managed System, the Flexible Service Processor (FSP), Logical Partitions (LPARs), and the Hardware Management Console (HMC).

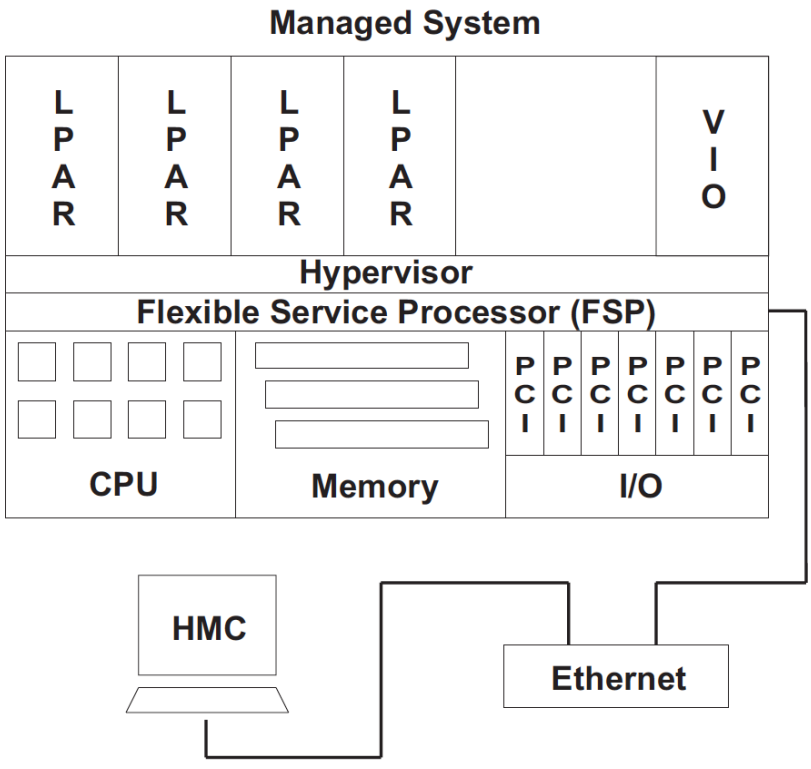

Figure 1

The Managed System

This is the major hardware component; what would perhaps more commonly be termed a server. This is the physical computer holding processors, memory and physical I/O devices. Managed systems can be broadly divided into three categories—small, midrange, and enterprise. They can also be classified based on the processor architecture. As of 2014, IBM is beginning to ship P8 systems (P8 designated POWER8, or the 8th generation of the POWER chip architecture released since its debut in 1990), however the majority of systems currently in production would be P7 and P6, and there are still more than a few P5 systems running.

Several general statements can be made about a managed system:

- All managed systems are complete servers, i.e. they have processors, memory, and I/O devices
- The number of processors varies depending on the system model. Small systems will typically have up to eight processors, midrange systems will scale up to sixty-four, and the enterprise P795 system (currently the largest) scales to 256 processors
- In any one managed system all processors will be the same architecture and speed, i.e. all 4.25 GHz P7 or all 4.2 GHz P8
- Like the number of processors, the number of memory slots also varies by model, as well as the capacity of the memory modules installed in those slots. Small servers might typically have a total of up to 64 GB memory, midrange servers up to 2 TB, and the P795 supports up to 16 TB of memory
- Midrange and enterprise class systems are designed to be scalable, hence a system can be ordered with a minimum amount of processors and memory and subsequently expanded by adding plug-in components up to the maximum capacity of the model of system; such expansion normally requires downtime to physically install the additional hardware
- In most cases systems have a fixed number of Peripheral Connect Interface (PCI) I/O device slots, the PCI version depending on the age of the server and which PCI variant was current at the time the server was introduced
- I/O capacity can be increased by adding I/O drawers containing either PCI slots, disk drive bays, or a combination of both slots and bays; these drawers are separately rack mounted from the server and connected using the Remote IO (RIO and RIO2) IBM proprietary loop bus architecture
- Most managed systems (the only exception being POWER blades, which are not very common) have a Flexible Service Processor (FSP), which is a key component in the virtualization architecture

Flexible Service Processor (FSP)

The FSP is a self-contained computer having its own dedicated processor, memory and I/O. It is located on the system board of all POWER servers (excepting POWER blades) and operates independently of the rest of the server. When the system power supply is connected to a power source (and before the server itself is powered on) the FSP is supplied power and boots up from code stored in NVRAM (nonvolatile real memory, or flash memory, although IBM uses the term NVRAM). This code is formally called system firmware, but is more commonly referred to as the Hypervisor. This term was coined when the first production use of virtualization was introduced by IBM on the System/360-67 in 1968. The hypervisor is the core software component responsible for mapping virtual processors, memory, and I/O devices to the actual physical resources of the server.

The FSP communicates to the outside world via an integrated Ethernet port. The IP address for the FSP can be supplied via DHCP (the default method), or it can be hard coded. If a web browser is pointed to the FSP IP address a simple graphical interface called the Advanced System Management Interface (ASMI) is provided. This requires a login ID and password unique to the ASMI, and is the method often used by IBM service personnel when performing service tasks such as upgrades or repairs. The FSP IP address is also used by the Hardware Management Console (HMC) to communicate with the managed system. We will talk about the HMC in more detail shortly.

Logical Partitions (LPARs)

The basic idea of server virtualization is to make one physical machine appear to be multiple independent machines. These imaginary servers are commonly called virtual machines (VMs), however IBM does not use this terminology, instead IBM uses the term Logical Partition (LPAR).

An LPAR requires processors, memory, and IO devices to be able to operate as an independent machine. It is the primary task of the hypervisor to allow LPARs access to the physical processor, memory, and I/O resources of the managed system in a controlled way determined by the desired configuration. Different LPARs will understandably have different resource requirements, so flexible resource management tools are important. The PowerVM environment offers a variety of configuration options to provide this needed flexibility.

In order to offer expanded IO configuration possibilities a special purpose LPAR called the Virtual IO Server (VIOS) is also a part of the architecture. We will detail how this fits in to the picture later in this paper.

Hardware Management Console (HMC)

The HMC is the central control point for virtualization operations on multiple managed systems. Physically an HMC is an Intel processor-based PC, most commonly mounted in the same rack as the POWER systems it manages. It runs Linux and hosts a Java-based application that forms the virtualization control point. It communicates via Ethernet with the FSPs of its managed systems using a proprietary protocol. The HMC is the only way to create and manage multiple LPARs on Power systems. It is possible to run a Power system without an HMC, but such a system can only operate as a single monolithic LPAR.

The HMC is not a point of failure in this architecture. While the data defining all LPARs across all managed systems managed by any one HMC is held on that HMC, the data for LPARs on any one system is also held in NVRAM by the FSP on that system. Consequently should the HMC fail, each system is able to continue operations using its locally stored data while the HMC is being repaired and/or recovered from its backup. Conversely, should any one system fail in such a way as to lose its local LPAR configuration data, after repair that data can be repopulated to the system from the HMC.

Resource Management

A key issue in any virtualization environment is the mechanism by which hardware resources are made available to virtual machines; or, using the IBM terminology, how the hypervisor distributes the managed systems resources among LPARs. The three basic hardware resources are computing capacity, real memory, and I/O.

Processor Resource Management

In the POWER environment an LPAR can be allocated one or more actual physical processors from the total number installed in the system. Such LPARs are termed Dedicated Processor LPARs.

An LPAR may also be allocated Virtual Processors (VPs). Within some quite flexible configuration boundaries, an arbitrary number of VPs can be allocated to an LPAR. Such an LPAR is formally termed a Micro Partition, although they are more commonly called Shared Processor LPARs (SPLPARs). Each VP in a micro partition appears to the operating system in the LPAR as a single physical processor; actual physical processor capacity in the form of time slices is allocated to VPs governed by a set of well-defined configuration parameters.

In either case, any one LPAR must have either exclusively dedicated or shared processors; the two processor types may not be mixed in a single LPAR.

Real Memory Resource Management

Memory is dealt with in similar fashion. A discrete amount of memory may be allocated to an LPAR from the total available in the system. Such an LPAR would be called a Dedicated Memory LPAR.

Alternatively a block of physical memory can be shared by multiple LPARs. In this shared memory model overcommitment is supported, e.g., a memory pool of 20 GB of memory could be shared by three LPARs, each of which has been allocated 10 GB of logical memory. In this case the operating system in each of the three LPARs thinks it has 10 GB of memory; in fact, there is only 20 GB to be shared between all three LPARs. Should the aggregate memory usage of the group of shared memory LPARs exceed the 20 GB of physical memory allotted, a system-level paging device is engaged to cover the overcommitment. This system-level paging is in addition to the normal paging device the LPAR must always have in any case, and is transparent to the operating system running in the LPAR.

As with processor configuration, any one LPAR may not mix these two types of memory allocation; all memory available to any one LPAR must either be dedicated to that LPAR, or drawn from a shared pool of memory. Additionally, LPARs using the shared-memory model are also required to be micro partitions, i.e., they must use the shared processor method of accessing processor resources.

I/O Resource Management

Managed systems all have some integrated IO devices (typically one or two disk controllers, anywhere from six to twelve internal disk bays, and possibly optical drive devices) and also additional device slots valuable to be populated with IO cards as desired. Details of the configuration of these devices vary from system to system.

In the PowerVM architecture IO resource is allocated to LPARs on a controller basis, rather than by individual device. Any LPAR can be directly allocated any integrated controller or PCI card slot, and therefore the card installed in that slot. Should that slot contain, for example, a four-port Ethernet card then all four ports now belong to that LPAR exclusively. It would not be possible to allocate the ports on the multi-port card individually to different LPARs. Similarly, if a disk controller, Small Computer Systems Interface (SCSI), or Fibre Channel (FC) supports multiple disks then all of those disks would be the exclusive property of the LPAR to which that controller was allocated.

This is potentially problematic as most power servers after P5 typically had more than enough processor and memory available to support more LPARs than there were IO slots available to service. This drove the development of the Virtual IO Server (VIOS), a special purpose LPAR that allows the creation of virtual disk and network controllers mapped to actual controllers on a many-to-one basis.

The Virtual IO Server (VIOS)

As the capabilities of physical network and disk controllers have increased in recent years, it has become possible for a single controller to meet the bandwidth requirements of more than one virtual machine. PowerVM makes use of the VIOS to leverage this capability. A VIOS is itself an LPAR. Physical Ethernet and disk controllers are directly allocated to the VIOS, allowing it to perform physical IO operations. Virtual controllers are then created in the VIOS and in client LPARs. Data IO is initiated by the client, flowing to the VIOS via the virtual controllers. The VIO then takes the virtual IO requests, converts then to real IO operations, performs the operations, and returns the data to the client. Let us examine how this works.

Disk IO Virtualization

A VIOS allocated a single disk controller supporting multiple disks controls all those disks, as discussed above. However, the VIOS can now allocate individual disks to separate virtual controllers mapped to different client LPARs, effectively allowing a single disk controller to be shared by multiple client LPARs.

Looking at Figure 2, we can see at the left that this managed system has three LPARs defined, all making use of the VIOS for IO. Each of the LPARs has at least one client virtual controller—those labeled vSCSIc are virtual SCSI clients, and those labeled vFCc are virtual Fibre Channel clients. Each of the client virtual controllers has a matching server virtual controller in the VIOS, shown as vSCSIs and vFCs. The managed system has an internal storage controller with three internal disks—A, B and C. This controller has been allocated to the VIOS, consequently the operating system in the VIOS sees those three disks, as indicated.

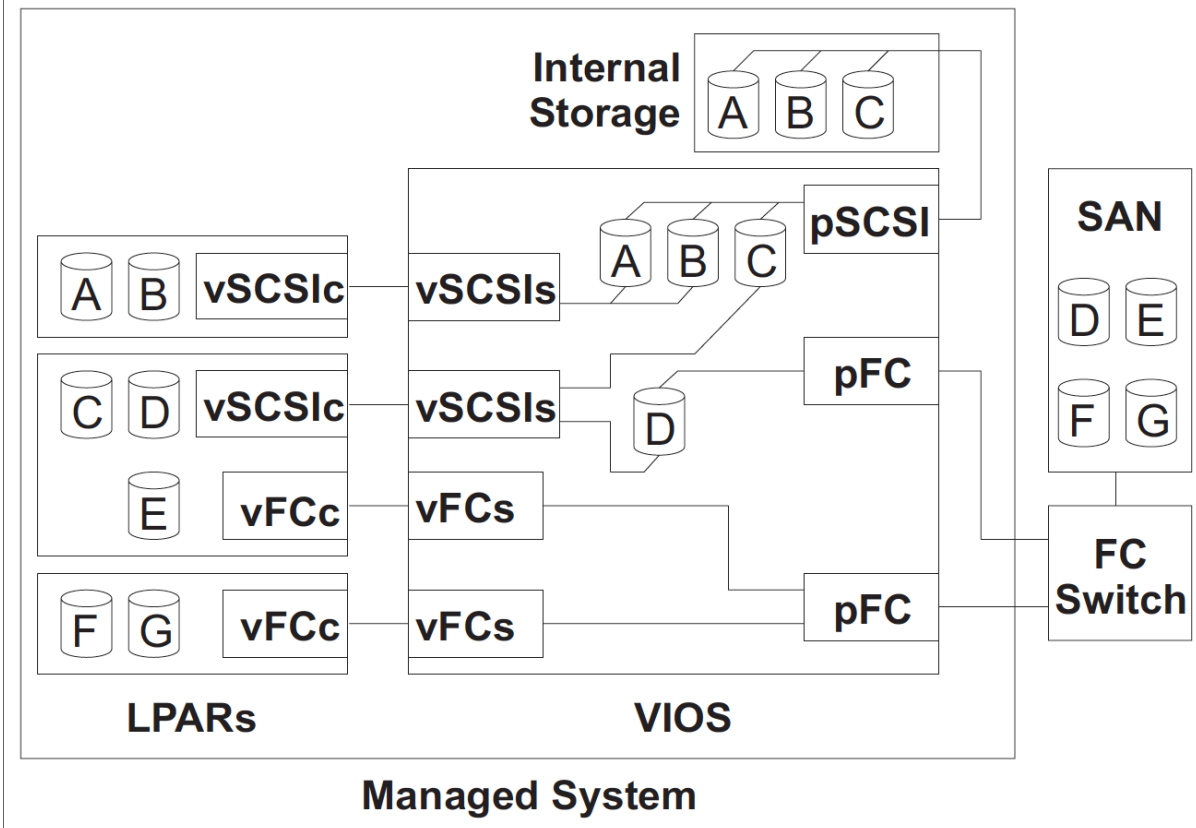

Figure 2

Now consider the LPAR at the top left. Two of the internal disks have been allocated to the server side virtual adapter for this LPAR, consequently the LPAR sees these disks. Although disk C is on the same physical controller, that disk has not been allocated to the server side virtual adapter; hence, the LPAR does not see it.

Next, consider the middle LPAR. Disk C has been allocated the server side virtual SCSI controller whose client is in the middle LPAR, so that is the LPAR that sees disk C. Additionally, the managed system has a physical FC controller (labeled pFC), and on the Storage Area Network (SAN) there is an array D that has been mapped to that pFC adapter. In this case as the array has been mapped to the pFC adapter, the operating system in the VIOS will see the disk, as shown. That disk could now be mapped to a vSCSIs adapter with the matching vSCSIc adapter in the middle LPAR, and that LPAR will now see disk D. Note that this LPAR will not know the difference between disk C and disk D. The LPAR will simply see two virtual SCSI disks, although in fact one is a physical disk local to the server and the other is actually a SAN hosted array.

Note now that the middle LPAR also has a virtual Fibre Channel client adapter (labeled vFCc). Virtual FC is a bit different than virtual SCSI. At the VIOS, the server side of a virtual FC adapter (labeled vFCs) is mapped not to disk devices but directly to a physical FC adapter (labeled pFC). This physical adapter, along with the physical FC switch it is connected to, must support a FC extension termed N-Port ID Virtualization (NPIV). In this case the vFCc

adapter in the LPAR is allocated a network identifier (called a worldwide port name, or WWPN) that is directly visible on the FC network, independent of the network identifier of the pFC card servicing it. The SAN administrator can now configure arrays mapped directly to a vFCc WWPN. These arrays (labeled E, F, and G in the example) are directly visible to the client vFCc adapters they are mapped to, and are not actually accessible by the VIOS itself; consequently, it is not necessary to perform a mapping operation at the VIOS to make an array visible to the client LPAR. Multiple vFCc adapters can be serviced by a single pFC adapter (as shown) hence the middle LPAR sees disk D, and the final LPAR in the example sees disks F and G as these three disks have been mapped by the SAN administrator to the WWPNs of the respective vFCc adapters.

Network IO Virtualization

Next let us examine VIOS virtualized networking, illustrated in Figure 3. In this figure, vEth represents a virtual Ethernet adapter configured in an LPAR. This adapter is implemented by the hypervisor, and is defined at the HMC as an element of the LPAR profile. To the operating system in the LPAR it appears to be a normal Ethernet adapter. These virtual Ethernet adapters are connected to a virtual switch, also implemented internal to the managed system by the hypervisor. This switch supports standard IEEE 802.1Q Virtual Networks (VLANs); each vEth adapter's port vLAN ID (as shown in the figure in the numbered boxes) is part of the definition of the adapter itself, and is assigned when the adapter is created.

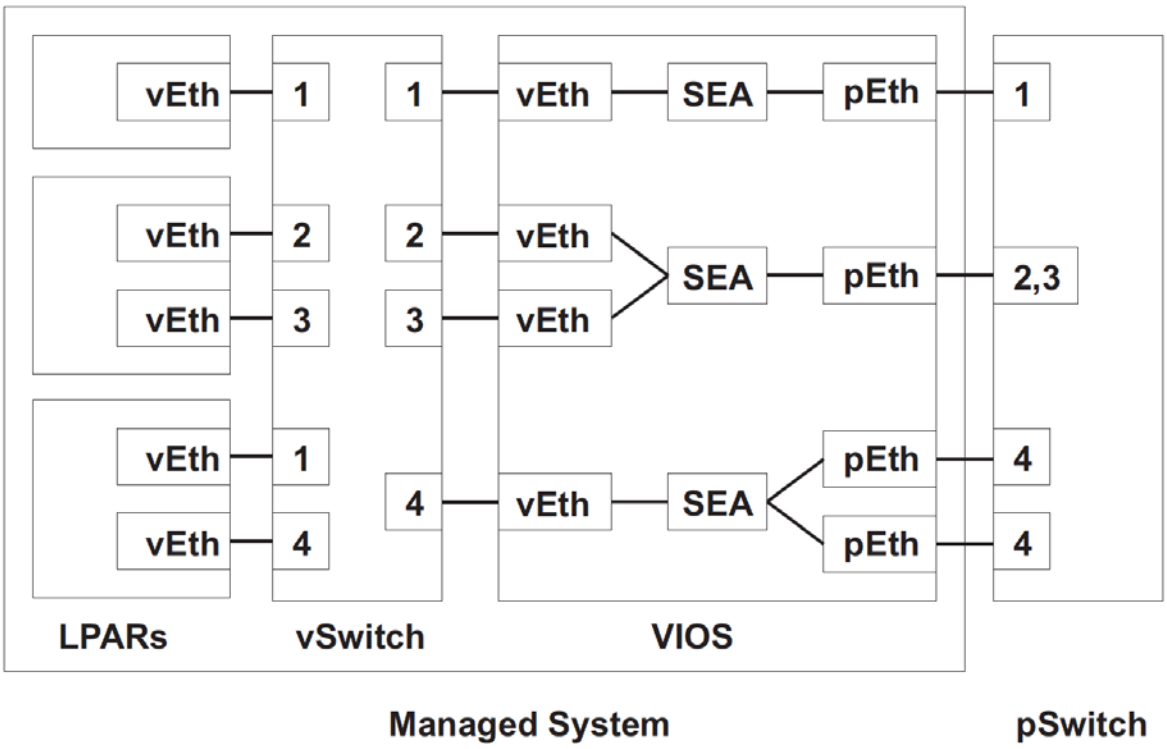

Figure 3

The VIOS also is configured with vEth adapters, thus LPARs can communicate to the VIOS using the internal vSwitch. The VIOS is also assigned physical Ethernet adapters (show as pEth in the figure). A layer 2 bridge is implemented in software in the VIOS to allow traffic to flow between the vEth and pEth adapters. Such a bridge is called a Shared Ethernet Adapter (SEA), as seen in the figure. An SEA may be configured in three ways: to bridge a single vEth to a single pEth; multiple vEth to a single pEth; or a single vEth to multiple pEth configuration. Each of these configurations has a purpose in terms of supporting multiple VLAN traffic, providing greater bandwidth, improving availability, or some combination of these three. As shown, any one VIOS may have several configured SEAs, as needs dictate.

In the figure, the LPAR at top left has a single vEth adapter on VLAN 1. As the vEth adapter in the VIOS, which is on VLAN 1, is configured as part of a single virtual to single real adapter SEA, all VLAN 1 traffic will pass through the top SEA. This is the simplest and also the recommended best practice configuration.

The middle LPAR at the left of the figure has 2 vEth adapters, each on a different VLAN. Because both of those VLANs (VLAN 2 and 3) are serviced by the same SEA, and because that SEA has a single pEth, all traffic for both of those VLANs will pass out through the middle SEA in the VIOS. If there were more LPARs with vEth adapters configured on VLAN 2 or 3, that traffic would also pass through the same SEA.

Finally, the bottom left LPAR also has 2 vEth adapters on different VLANs, but traffic from that LPAR will end up going through different SEAs due to the VLAN configuration. The VLAN 4 traffic SEA is configured with multiple pEth adapters. These would be configured as a link aggregation in order to increase bandwidth and availability.

Note that unlike the configuration for virtual disk IO, the virtual network client and server configuration are independent of each other. Once SEAs are in place to service the necessary VLANs, any LPAR can now be configured with a vEth adapter on the required VLAN. It is not necessary to create matching client/server adapter pairs as it is for SCSI or FC disk virtualization. Also, in this overview we have shown only a single VLAN per each vEth adapter; in fact, a single vEth adapter can service multiple VLANs as long as the proper operating system network configuration is in place.

IO Virtualization Redundancy

So far all of our examples show a single VIOS. Clearly this would be a significant point of failure should all LPAR IO rely on a single device. In fact, the likelihood of a VIOS failure is actually low. A failure of the managed system would imply VIOS failure, but also all clients would be likewise affected, and the only way to address that would be to have two or more managed systems operating as an availability cluster, which is in fact how the failure case of managed system loss is handled. There are different ways in which this can be done, which are beyond the scope of this paper.

The more significant issue is maintenance. The VIOS operating system is in fact a modified version of AIX, which, like all operating systems requires patching and upgrading on an ongoing basis. Also like most operating systems it is reasonable to expect that some downtime is going to be incurred as a part of the upgrade/patch process. If that implies a need to restart the VIOS, then presumably that would translate into a restart requirement for all clients, if there were only one VIOS. In order to avoid this, the normal recommended best practice is to implement a dual redundant VIOS configuration, as show in Figure 4.

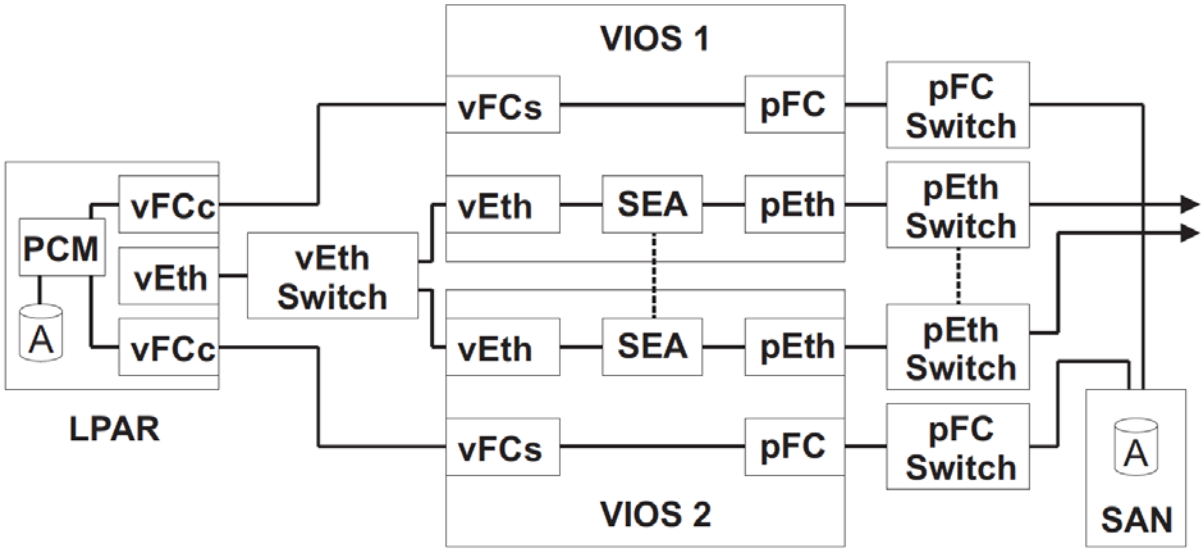

Figure 4

At the left is a typical LPAR utilizing the VIOS for both network and disk IO. It requires only one vEth adapter, which allows it a connection to the internal virtual Ethernet switch. Both VIOS have a connection to this switch, allowing two independent, redundant paths to the external physical network infrastructure. The dotted line between the two SEAs implies that they communicate with each other in order to determine how traffic will be managed when either or both of the VIOs are running. This management is automated based on configuration, so if a functional network path through either VIO is lost, failover is automatic and fast (typically < 5 ms, as long as the external network switch routing tables are also synchronized, indicated by the dotted line between the two physical switches).

In terms of disk IO, the LPAR is configured with two virtual disk controllers, each pointed to a different VIOS. The example shows the use of vFC, vSCSI is also supported in a dual VIO configuration. As multiple adapters introduce multiple paths to the same SAN array, a path control module (labelled PCM, in the client LPAR) is necessary to decide which path will be in use based on the status of the VIO servers, as well as the FC and SAN infrastructure. In the case of vFC, the PCM is typically implemented in the LPAR, if vSCSI is used the PCM is implemented in the VIOS. It is also possible to use multiple physical adapters in the VIOS to increase the level of physical path redundancy, and vFC and vSCSI can both be used in the same configuration, so this picture can look quite different in any given actual site configuration.

Conclusion

In this paper we have provided an overview of the PowerVM virtualization platform architecture. There are also a number of further PowerVM related capabilities we have not discussed including (but not limited to) Dynamic Logical Partitioning (DLPAR), which allows the resource configuration of LPARs to be altered while the LPAR is in live production; Live Partition Mobility (LPM), which allows an LPAR to be moved between managed systems while in full production and without disruption; and Active Memory Expansion (AME), allowing the substantial processing capacity of Power7 and Power8 processors to be brought to bear to compress memory stored data in order to reduce the real memory footprint for any one LPAR.

Actual systems require expert knowledge of the various configuration parameter details necessary to obtain the optimum result in a production environment. IBM certified training courses are an excellent place to acquire the necessary skills.

Learn More

Learn more about how you can improve productivity, enhance efficiency, and sharpen your competitive edge through training.

- Power Systems for AIX I: LPAR Configuration and Planning (AN11G)
- Power Systems for AIX - Virtualization I: Implementing Virtualization (AN30G)
- Power Systems for AIX - Virtualization II: Advanced PowerVM and Performance (AN31G)

Visit **www.globalknowledge.com** or call **1-800-COURSES (1-800-268-7737)** to speak with a Global Knowledge training advisor.

About the Author

Iain Campbell is a mechanical engineer by profession. While responsible for managing several production automation labs at Ryerson Polytechnic University in Toronto, he became distracted by UNIX operating systems. His first experience of AIX was a PC RT used in the lab as a machine cell controller. Iain has been teaching and consulting in AIX and Linux since 1997. He is the author of *Reliable Linux* (Wiley, New York, 2002), as well as several technical papers and curriculum material. He holds LPI and Novell certifications in Linux administration, and is an IBM certified AIX Specialist.