

A White Paper from Neterion

Hyperframe™

Enabling Virtualization in the Datacenter

Introducing Neterion's Xframe® V-NIC™ family
with Hyperframe™ I/O Virtualization Technology

January 2007

Introduction

Virtualization is a key strategy for simplifying deployment of IT resources and maximizing their utilization. Specifically, virtualization refers to the concept of abstracting physical resources such as compute cycles, data storage, and network bandwidth, and then provisioning and sharing these resources amongst multiple applications. For example, a single server may be “virtualized” to allow multiple operating system (OS) images to run concurrently; the amount of storage available to a user on a Storage Area Network (SAN) may be dynamically adjusted on the fly; and the amount of bandwidth allocated to a given application may be boosted or reduced as required. Virtualization began as a niche market but is rapidly gaining acceptance as the preferred way to manage and provision system resources within a network.

The benefits of virtualization are well understood. System administrators are able to capture underutilized resources and re-allocate them to constrained applications. Resources can be dynamically allocated and load-balanced as the characteristics of traffic and applications change over time. Hardware can be transparently replaced or upgraded with a minimum of downtime. Utilizing existing resources more efficiently in this way leads to reduced infrastructure cost, better utilization of IT assets, lower power consumption, reduced cooling requirements, and inevitably lower total cost of ownership.

Today, server virtualization is primarily handled through specialized software provided by vendors such as VMware. The success of this technology has led system architects to think about ways to increase virtualization’s effectiveness by extending the same concepts to the hardware level. Companies such as Intel, with its Virtual Technology (a.k.a. Vanderpool), and AMD, with “Pacifica”, are implementing virtualization-specific features in their CPUs. Following along the same path, input/output (I/O) architectures are now being redesigned to support this powerful concept from end to end, with the introduction of hardware-based I/O virtualization, commonly referred to as **IOV**.

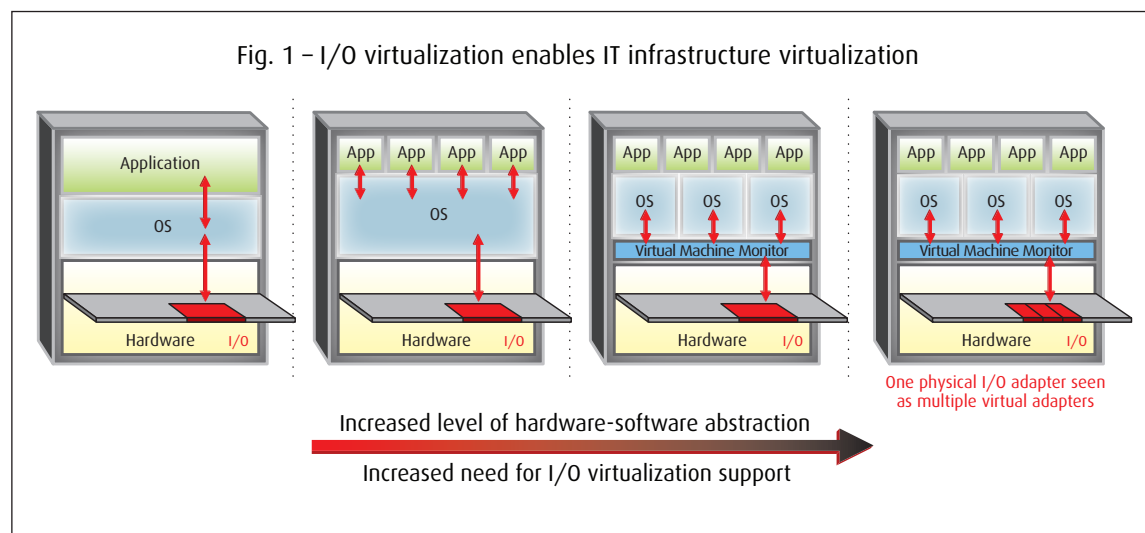
Neterion’s **Hyperframe™** IOV architecture offers the industry’s first comprehensive set of I/O virtualization features implemented at the hardware level. Neterion’s **Xframe®** line of 10 Gigabit Ethernet adapters is built on the principles of the Hyperframe architecture. It is the only network interface designed from the ground up for I/O virtualization — we call it the Xframe **V-NIC™**.

The Need for IOV

The basic principle of IOV is to share an I/O (or pool of I/O) sub-systems among various compute systems — different images of an Operating System in a server, or different blades in a chassis, each running one or multiple images of an OS. An example of an I/O sub-system is a network interface adapter. Implementing an IOV architecture for the network interface means abstracting one or several adapters and making them appear like a multitude of independent interfaces to the virtualized compute systems.

One of the primary benefits of IOV is the reduction in Total Cost of Ownership (TCO). IOV enables consolidation of I/O resources, which both reduces equipment and implementation costs, while allowing for better centralized and more efficient network management.

As an example, in today’s architecture, a virtualized server running VMware, for example, would typically host multiple network adapters, each assigned to an OS image or an application. Using IOV technology, it becomes possible to replace multiple gigabit adapters with one physical 10 Gigabit Ethernet adapter, viewed as separate logical adapters (or “virtual adapters”). This virtualization at the I/O level results in higher flexibility for IT managers, who can allocate dynamically the overall 10 Gigabit Ethernet bandwidth across the various OS instances and applications, instead of being confined to a fixed Gigabit for each. The bandwidth allocation process can follow a prioritization and Quality-of-Service algorithm (for example by line of business, or time of day, etc.), further enhancing the experience for the end-users of these applications. Finally, as IT managers reduce the number of physical devices they manage by a factor of 10 in a virtualized environment, the administration complexity (and costs) are greatly reduced as well.



IOV also ensures system reliability through hardware isolation. Isolation is critical from a protection perspective in that one application or operating system image cannot adversely affect the performance or reliability of another. One of virtualization’s primary strengths is the fact that network components are unaware that their access to system resources has been virtualized. For example, an OS image assumes that it is the sole owner of an output port even though this port may be shared among several OS images across multiple CPUs or blades. To be truly isolated each OS image must be able to treat each system resource as if the resource belongs only to that OS image, even though these resources are actually shared. Such complete isolation is essential for easing deployment and management of new applications.

Finally, for all practical purposes, it is always possible to double-up the virtualized interface. In our example, a second 10 Gigabit Ethernet port would be present in the system, to provide for redundancy and fail-over capabilities, or to double the network pipe (20 Gigabit), or a combination of both (see Table 1).

Table 1 – Primary Value Proposition of IOV

- **Cost:** replace ten 1 Gigabit interfaces with a single 10 Gigabit adapter, reducing complexity and administration costs
- **Flexibility:** dynamically allocate bandwidth across the various components of the system (different OS images or different blades in a chassis, or both), instead of limiting the bandwidth to a fixed amount per component
- **Performance:** a hardware-based IOV architecture reduces the processor-intensive software layers required to pool or trunk the Gigabit interfaces together (the only method available in absence of hardware IOV)
- **Isolation:** every OS image is presented with its own independent I/O path; it is completely unaware that it is sharing resources with other OS images, and cannot harm them by virtue of true hardware protection
- **Reliability:** with a second 10 Gigabit interface in the system, users can implement redundancy and fail-over capabilities, or double the size of the network pipe (20 Gigabit total) to share among the components of the system. Optionally, a combination of fail-over and higher bandwidth, all dynamically managed and “virtualized”, is achievable

IOV Today

IOV has been difficult to implement because of a general lack of hardware support for virtualization. While companies like VMware have done excellent work with regards to support for I/O virtualization, a software-only implementation imposes a significant burden on the processor at multi-Gigabit network speeds. For example, software has to decide which virtual OS image a received packet should be directed to at rates that potentially exceed 15 million packets per second.

Although software virtualization offers outstanding value to the end-user customers, it has a big appetite for compute cycles that could otherwise be used to execute applications. Abstracting hardware at a high level in software is costly in terms of CPU capacity, requiring many more cycles than an equivalent non-virtualized system. When soft virtualization is married with non-virtualized 10 Gbps Ethernet, high performance is difficult to achieve.

Fortunately, 10 Gigabit Ethernet adapters like Neterion's Xframe® V-NIC™ are available today, and with the Hyperframe™ I/O virtualization technology are well ahead of the deployment curve. Designed with the IOV framework in mind, they already offer full hardware support for I/O virtualization.

The Xframe V-NIC product family offers a unique multi-channel device model. A total of eight independent, hardware-based transmit and receive paths are available and each path may be prioritized for true Quality-of-Service support. By combining this novel architecture with the power of Extended Message Signaled Interrupts (MSI-X), Neterion has established a leading-edge framework for I/O virtualization (IOV). This set of unique, industry-leading IOV features is what constitutes Neterion's Hyperframe technology.

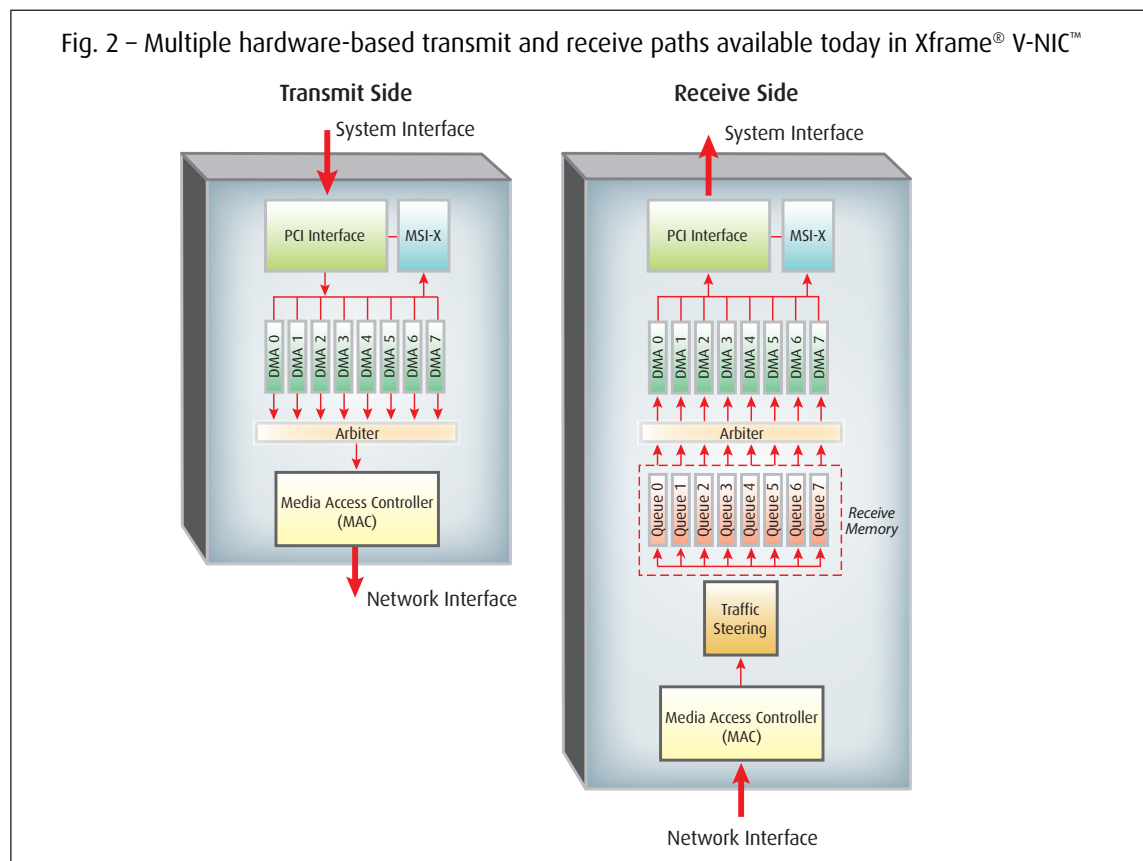
Other vendors will claim thousands (or better, tens of thousands) of virtual paths. But beware that these software-only implementations are very limited in nature. Neterion is the only manufacturer to support these IOV features in hardware, as an integral part of the Xframe V-NIC design philosophy that implements the Hyperframe architecture.

With the combination of IOV-based network network interfaces such as Neterion's Xframe V-NIC adapters with Hyperframe technology and versions of virtual OS environments like VMware that are optimized for IOV, users will soon be able to take advantage of end-to-end virtualized systems, extending the benefits of virtualization at the I/O level.

IOV Capabilities

Neterion’s current V-NIC adapters support the following IOV features. This set of unique, industry-leading features is what constitutes Neterion’s **Hyperframe** technology:

- **Multiple receive and transmit queues physically separated in hardware** (see Figure 2):
 On the transmit side, having multiple, separate transmit queues allows several entities in the system to have unique and dedicated access to the transmit path. Jobs from one process will not interfere with those from another, and hardware takes care of arbitrating for the link and retrieving data from the host system. OS images are unaware that they are sharing the link. On the receive side, having multiple queues means that traffic is sorted and prioritized by hardware. This relieves the software from having to classify and steer frames to the appropriate OS image, resulting in significant CPU utilization savings. Traffic is distributed among the receive queues by means of a sophisticated receive traffic classifier. In addition, it may also eliminate extra memory copies on the receive side.



- **Independent DMA hardware engines**
 Having multiple DMA engines allows the adapter to maintain a completely separate context for each receive queue, allowing for isolation of OS images. Specifically, an individual OS image can maintain its own set of receive frame resources on a per-engine basis. On the transmit side, up to eight fully independent transmit operations may be underway at any time, providing critical “non-blocking” capabilities that drive bus utilization since the adapter does not have to wait for the data from DMA to be read or written to host memory in order to proceed with handling data from another DMA engine.

- **Separate MAC addresses**

Xframe adapters may be configured with hundreds of unique unicast/multicast addresses, which allow a single adapter to appear as multiple adapters to the network. This also simplifies routing of packets to the appropriate OS images. When multiple OS images are running on a single CPU, it becomes more difficult for a system to manage “independent” links that share the same MAC address, introducing switching overhead and contention. With individual MAC addresses, each link can truly be independent.

- **Independent interrupts**

Xframe supports MSI-X (eXtended Message Signaled Interrupts), an industry standard, PCI-compliant interrupt mechanism. Most adapters support only INTA, which utilizes a physical pin. Interrupts are the primary mechanism for an adapter to alert the system that they require attention. With INTA, when the adapter wants to interrupt the system (ie, a packet has arrived), no matter what the reason, all it can do is assert a single physical interrupt. INTA is unable to distribute the interrupt, so a single entity must handle all the interrupts for the adapter. In the MSI-X scheme, Xframe can generate up to 64 unique interrupts, which allows each path to signal the system with its own unique interrupt. This permits the adapter to independently notify only the appropriate OS image that traffic is available for it without interrupting all the others.

- **Contextual interrupts**

With INTA, the adapter is only capable of generating a single hardware interrupt for the entire system, and the driver has to read the adapter’s interrupt reason register in order to determine why an interrupt has occurred. This is a very expensive process in terms of CPU utilization. With MSI-X, when the adapter performs an MSI-X write, the target OS image knows exactly why it was interrupted — because there are frames ready for it to service — and can more quickly determine what is required of it.

- **Independent interrupt moderation schemes**

Interrupt moderation allows the adapter to dynamically adjust the way it interrupts the system based on the traffic flow on a given virtual channel. Consider the impact of an architecture that does not support independent interrupt moderation schemes. Effectively, each channel — regardless of application, data type, or service-level agreement — will be forced to use the same moderation scheme, which means that most links will be operating at less than optimal performance and quality of service. An adapter that supports independent interrupt moderation schemes enables IT administrators to select the appropriate scheme for each context, allowing the adapter to optimize its operation for throughput, latency or quality of service as required. For example, if a given receive queue is receiving a lot of data, the adapter can interrupt less frequently in order to allow the OS image to service the link in more efficient bursts. However, if a given receive queue is receiving a small amount of data, the adapter can minimize latency by interrupting the appropriate virtual instance more frequently. Up to 72 separate sets of interrupt moderation parameters are available (64 on transmit, 8 on receive). On the transmit side, any of the 64 interrupt moderation schemes may be assigned to any of the virtual paths. On the receive side, there is one moderation scheme available per virtual path.

OS vendors, like VMware, have already begun extensive work to add hardware-based I/O virtualization support. For example, implementation of “passthru” models — allowing the adapter’s hardware channels to be mapped transparently to different OS images or applications — will greatly optimize the utilization of the IOV features available by some network vendors.

On the one hand, “passthru” models provide much better support for high-speed IOV, but on the other hand, they require hardware to pick up the features that are being removed from the OS.

In that respect, Neterion’s Xframe V-NIC is absolutely unique in its ability to support these emerging models. With Xframe V-NIC and Hyperframe technology, multi-path communication is entirely handled in hardware, inside the ASIC, providing for superior performance and true hardware isolation and protection between channels.

IOV Tomorrow

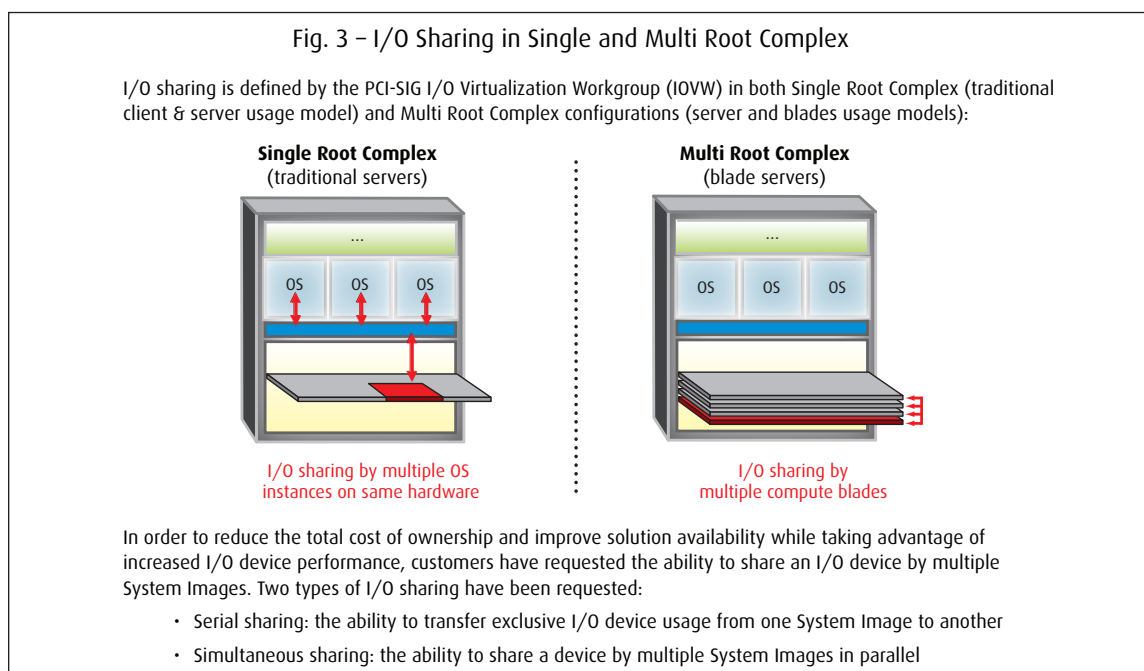
In addition to working closely with leading virtualization vendors like VMware to include Hyperframe support in their OS environment, Neterion is an active member of the I/O Virtualization (IOV) working group of the PCI-SIG.

The PCI-SIG IOV group defines two levels of IOV configurations: Single Root Complex and Multi Root Complex (see Figure 3). Single Root Complex refers to traditional servers, where I/O is shared by multiple OS images on the same hardware. An application of Multi Root Complex will be in blade computing, where I/O is shared by multiple hardware instances. Each present different implementation challenges, especially when both are deployed at the same time.

For an adapter to be able to robustly support I/O virtualization for single and multi root complex configurations, many mechanisms need to be in place to enable true independent operation of each virtual adapter link. This means that each channel must have its own hardware resources independent from every other channel, including:

- Separate receive and transmit queues
- Independent DMA engines
- Separate MAC addresses
- The ability to classify and steer receive traffic
- Separate interrupts
- Independent interrupt moderation schemes
- Separate register sets
- Separate copies of configuration space
- Ability to direct packets through the PCI-Express fabric

With Hyperframe and Xframe V-NIC, Neterion already supports most of these features today, and therefore offers the broadest IOV support in the Ethernet adapter market. With its next-generation Xframe V-NIC products, it will support the entire list, making efficient and robust IOV both affordable and available off-the-shelf.



PCI-Express and I/O Virtualization

The I/O Virtualization (IOV) Workgroup within the PCI-SIG has been charged with amending the PCI specification to enable virtualization. Specifically, it will amend PCI-Express, which is the proposed bus for most applications targeting I/O virtualization. The workgroup is made up of major industry players including AMD, ATI, Broadcom, Emulex, HP, IBM, IDT, Intel, LSI Logic, Microsoft, Neterion, NextIO, Nvidia, PLX, QLogic, Stargen, Sun, and VMware. The target date for completion of the spec is November of 2006.

Two alternatives to PCI-Express were initially considered as potential IOV fabrics: InfiniBand, a low-latency interconnect technology, and the Advanced Switching Interconnect (ASI) architecture. Neither of these alternatives, however, was as well suited to the system as a whole as PCI-Express is, from a functional, management, and cost perspective.

InfiniBand, for its part, has the methods, protocols, and management mechanisms required to share I/O devices across a fabric. Although still recent from a deployment perspective, this technology has been used to demonstrate the benefits of I/O sharing and virtualization. However, while InfiniBand could be used to virtualize server resources, another technology would have to be used for clients, which would lead to the creation of a heterogeneous network. This would increase management and deployment complexity. Additionally, InfiniBand is targeted at large-scale shared fabrics — while it can be scaled down into very simple configurations, the requisite functionality goes well beyond the needs of other market segments, unnecessarily burdening developers and increasing cost. Other difficulties with InfiniBand include its message-based mechanisms that require native PCI-Express transactions to be encapsulated, and its general lack of support for non-network applications such as graphics. While encapsulating PCI-Express over InfiniBand can enable interoperability, the added cost of providing translation bridges to and from InfiniBand does not align with the cost constraints associated with servicing multiple market segments.

The other alternative, Advanced Switching Interconnect (ASI), was defined to provide the methods, protocols, and management mechanisms required to tunnel a variety of protocols across a switch-based fabric, including the ability to transport PCI-Express via an ASI-to-PCI-Express bridge function. However, ASI is not so much a virtualization technology as it is a routing/crossbar protocol.

The IOV workgroup has addressed the shortcomings of InfiniBand by recommending that IOV focus on small fabric diameter topologies, making it applicable and efficient to implement across the network. Additionally, IOV is slated to support all types of I/O, including graphics applications. From an ASI perspective, the view of this group is that the industry and its customers are best served by extending the existing PCI-Express specification suite to natively support IOV technology.

In general, the targeted market segments and usage models align directly with the cost constraints and complexity already taken into account by the PCI-SIG specifications. Being packet based, PCI-Express lends itself well to virtualization since packets can be easily directed to the appropriate network entity. PCI-Express also has the notion of switches already built into it. Therefore it makes the most sense to extend the specifications rather than increase system complexity by introducing new specifications and protocol layers.

Conclusion

Hardware-based I/O virtualization brings new levels of efficiency to existing virtualized server environments while achieving cost savings in acquisition, administration, power, and cooling. IOV not only enables lower infrastructure cost and higher performance, it changes the fundamental way in which I/O interacts with systems. With Neterion's **Hyperframe™** IOV architecture supported in the **Xframe® V-NIC™** line of 10 Gigabit Ethernet adapters, I/O promises to be as flexible and versatile as any other component in the virtualized chain. Revolutionary new types of compute architectures will emerge. A shared and virtualized I/O model will offer a cost-effective and secure alternative to the traditional model that requires dedicated Ethernet ports for each OS instance.

Hardware-based IOV is a logical extension to both OS virtualization environments and the recent hardware enhancements that bring virtualization support to the CPU level. Together with these components, IOV will soon provide an important piece of the ecosystem, enabling the vision that vendors have been promising and that end-users have been waiting for: a true virtualized datacenter.

About Neterion, Inc.

Founded in 2001, Neterion Inc. has locations in Cupertino, California and Ottawa, Canada. Neterion delivers 10 Gigabit Ethernet hardware & software solutions that solve customers' high-end networking problems. The Xframe® line of products is based on Neterion-developed technologies that deliver new levels of performance, availability and reliability in the datacenter. Xframe, Xframe II and Xframe E include full IPv4 and IPv6 support, and comprehensive stateless offloads that preserve the integrity of current TCP/IP implementations without "breaking the stack." Xframe drivers are available for all major Operating Systems, including Microsoft Windows, Linux, Hewlett-Packard's HP-UX, IBM's AIX, Sun's Solaris and SGI's Irix.

Further information on the Xframe V-NIC technology can be found at <http://www.v-nic.com/>