Architectural Directions for Server I/O Subsystems

Dr. H. Pat Artis
Performance Associates, Inc.
Pagosa Springs, CO 81147
drpat@perfassoc.com

Abstract: From the perspective of an experienced performance analyst, perhaps the most frustrating aspects of server I/O are the lack of hardware measurement data and the architectural limitations imposed by the traditional bus structures. This paper is intended to provide a tutorial on IBA, the InfiniBand Architecture. It will discuss what the IBA will mean for the design of enterprise class servers, the generic layered driver model, as well as storage area networks in the future. Specific emphasis is placed on the primitives for I/O measurement that are incorporated in the architecture.

1. Introduction

While the broad range of servers (i.e., WINTEL, Sun, et al) available in today's marketplace provide a wide range of capabilities for enterprise class computing, they share a number of common I/O related limitations. These limitations have motivated the development of new architectural standards for the design and development of future servers.

During 1997 and 1998, two competing consortiums were formed to specify a new I/O architecture for servers. These groups were Next Generation I/O (NGIO) and Future I/O. The NGIO consortium was headed by Intel and included SUN, Dell, Hitachi, NEC, and Siemens. Disappointed that they were not invited to participate in the specification of this new architecture, Compaq, IBM, HP, 3COM, and Adaptec formed Future I/O. Fortunately, sanity prevailed and the two groups merged during late 1999 to form the InfiniBand Trade Association[1] (IBTA). The IBTA is managed by a group of steering directors Compaq, Dell, Hewlett-Packard, IBM, Intel, Microsoft and Sun Microsystems.

The IBTA has three primary objectives:

- First, the organization has developed a specification that will meet the emerging bandwidth requirements of server solutions. Channel based, switched fabric architecture will deliver scalable performance to meet the growing demands of data centers; flexibility to provide connectivity that scales with a business' demands, independent of the microprocessor or OS complex; and flexibility to inter-operate from the entry level to the enterprise,

- Second, the architecture draws on existing proven technology. Switched-fabric, point-to-point interconnects are not new to the industry. InfiniBand Architecture will utilize the collective knowledge of switched fabric implementations to deliver the best and most cost-effective I/O solutions, which eventually ensures a transition from legacy I/O like PCI and PCI-X, and

- Third, it employs a governance model that effectively balances the need to drive the technology forward quickly and, at the same time, involves the industry throughout the development process.

[1] The group was briefly called System I/O during its formative period. Today, more than two hundred hardware and software concerns are members of the IBTA. See www.infinibandta.org .

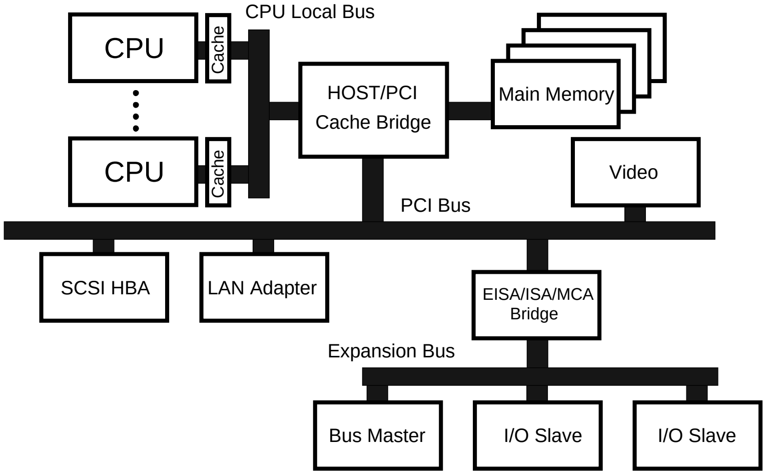

Figure 1. PCI Bus Architecture

Version 1.0 of the InfiniBand specification was introduced in October of 2000. [1] While initial product deliveries are expected in the second half of 2001, a broad range of products should become available in 2002.

The primary intent of this paper is to introduce the I/O measurement primitives that are part of the General Services Interface. Based on this introduction, the author will speculate about the high-level measurement tools that can be developed based on the InfiniBand Architecture (IBA) I/O measurement primitives.

2. Limitations of the Existing Bus I/O Model

The existing bus I/O model for servers presents us with a number of significant performance and scalability issues. While a complete list of them is beyond the scope of this paper, there are three main architectural characteristics that relate to the overall capacity, performance, and integrity of current servers. They are:

- Physical integration of the server's processor, memory, and I/O resources on a backbone bus,
- Inherent limitations of bus architectures, specifically the limitations of bus data rate, number of interface slots, and the bandwidth available to the slots, and
- Potential operating system integrity exposures as well as the resource requirements of ***kernel mode*** host-bus adapters (HBA) drivers.

While not directly related to the overall capacity, performance, or integrity of current servers, it is important to note that the existing bus I/O model does not incorporate measurement facilities.

Since the first two issues are common problems for all of today's bus architectures (e.g., PCI and Sun S-BUS), we will employ the 64-bit 66 MHz PCI bus and the proposed PCI-X bus specification as examples to discuss them. Figure 1 provides an overview of the structure of the PCI Bus. [2]

Version 2.1 of the PCI bus, released 1Q95, provides physical connectivity between an SMP (symmetric multiprocessing) local bus processor resource, main memory, PCI cards (e.g., SCSI and LAN adapters), video, as well as support for prior generation ISA, EISA, and perhaps MCA cards.

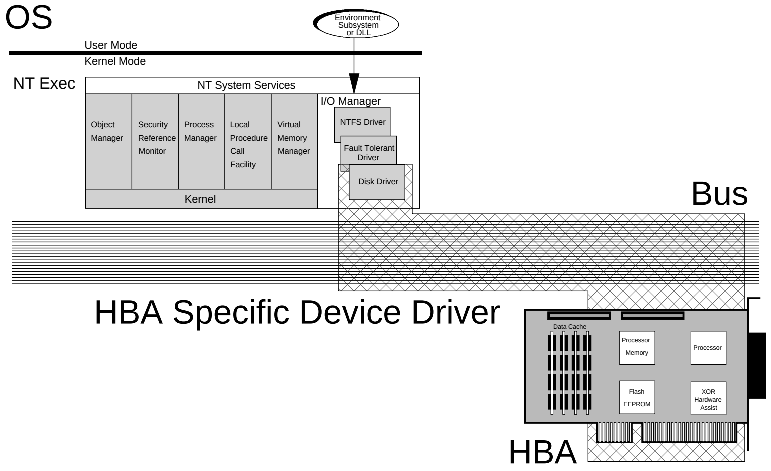

Figure 2. Windows NT Layered Driver Model

To conserve physical bus slots, the video and LAN adapters have been incorporated into the majority of high-end server motherboards. As a shared backbone resource, this bus defines the ultimate bandwidth that the server platform can provide. In addition, the design is subject to a problem known as ***slot saturation***. That is, when all of the slots have been occupied, you have defined the maximum physical connectivity for the server. While it is natural to ask ***why not just add more slots to the bus?***, busses suffer from the same signal skewing restrictions as parallel cables. *The longer you make a bus or parallel cable, the lower its maximum aggregate data rate!*

In an attempt to provide a growth path for the PCI architecture, IBM, HP, and Compaq formed the PCI-X working group to develop an extension to the PCI bus architecture to support fibre channel, gigabit Ethernet, and Ultra-3 SCSI. [3] While the PCI-X bus specification doubled the 66 MHz speed of the PCI bus, it could only support a single slot at the 133 MHz rate. When more than one slot was configured, the bus rate dropped to 100 MHz. Even when configured with just one slot, the PCI-X bus only provides an aggregate bandwidth of approximately 1 GB/sec

The third architectural issue was the software integrity exposures and resource requirements introduced by the low-level HBA hardware drivers in the operating system's layered driver model.[2] An overview of the Windows NT layered driver model and its relationship to the HBA are shown in Figure 2.

In the Windows NT layered driver model, the driver stack is comprised of the NT File System (NTFS) driver, the fault tolerant driver, and an HBA specific driver that intercedes between the operating system and the host bus adapter. The HBA specific driver (crosshatched area in the figure) executes in the NT kernel and communicates over the bus to control the operations of the HBA. That is, there is not a layer of hardware abstraction between the operating system and the third-party vendor supplied code that controls the HBA.

In a perfect world, all drivers would be error free and there would not be any conflicts between different drivers installed on a server. Unfortunately, the world is far from perfect. Hence, most NT system administrators have experienced the feared

[2] Please note that this discussion is not a criticism of the layered driver model. Rather, it is intended to discuss the security exposures introduced by the inclusion of third party hardware drivers in the kernel of the operating system.

blue ***screen of death*** after system maintenance. While the specific symptoms are different in the UNIX variants (e.g., SOLARIS or AIX), the fundamental problem remains the same.

As an interesting aside, Novell has addressed the software integrity issue by requiring that all HBA vendors submit their drivers to Novell for certification. While this process improved system integrity, it has also become a significant burden for Novell. As a result of this experience, Novell has been a staunch supporter of the Intelligent I/O (I_2O) initiative. [4] Briefly, the I_2O architecture defined a dedicated channel resource (incorporated in an HBA) in which the vendor specific driver operates. The vendor specific HBA driver receives logical requests from a standard operating system facility in the layered driver model. The benefit of this approach is that no third-party code need be introduced into the operating system's kernel to support the I/O process. The IBA incorporates and expands on the I_2O initiative by moving the concept from a single HBA based channel card to a switched channel subsystem resource.

In addition to Novell, Windows 2000 supports the I_2O processing model. While a ground breaking and interesting initiative, InfiniBand incorporates and vastly expands the channel concept defined by I_2O.

3. InfiniBand Architecture Overview

While the objective of the PCI-X bus was to address interim bandwidth problems, its real benefit was the conclusive demonstration that faster busses were not going to solve the I/O bandwidth problem. Rather, it was clear that switched fabric should be employed to connect the systems resources. Since switched serial connections (e.g., fibre channel) are far less subject to distance restrictions than bus architectures, it became obvious that the HBA resources (i.e., I/O subsystem) need not be directly incorporated in the server. Rather, the HBAs could be supported by specialized I/O processors like those demonstrated by I_2O rather than depending on the server in any manner for processing resources. Moreover, the existing I_2O implementation had demonstrated both the resource consumption and software integrity benefits of isolating the HBA specific drivers on their own specialized processors linked by a transport layer to a logical driver within the operating system. While it is substantially greater in scope, *the InfiniBand architecture provides the same hardware abstraction and scalability features for open systems as the System/370 External Data Controller (EXDC) did for MVS/XA in the early 1980s.*

A detailed discussion of the InfiniBand architecture is far beyond the scope of this paper since the two volumes of Version 1 of the IBA specification [1] exceed 1,500 pages. This paper will provide a high level overview of three areas of the specification. They are:

- Driver isolation,
- Host and target channel adapter model, and
- Subnets and physical connections.

The reader should note that while this discussion provides the basis for discussing the measurement primitives incorporated in the architecture, it is far from even being a meaningful high-level overview of the IBA.

Figure 3 provides an overview of how third party drivers are isolated in IBA I/O cards. It can be directly compared with the discussion of the NT layered driver model previously shown in Figure 2. The top half of the figure represents a hypothetical[3] version of Windows NT designed to exploit the IBA. The left-hatched box at the bottom of the layered driver model is an operating system provided communication driver. The operating system[4] issues an I/O request through a standard protocol engine that is passed through the network to an I/O card, i.e., channel. The request is then executed by a dedicated microprocessor resource incorporated in the I/O card.

[3] Please note that the development of logical drivers for Windows NT and/or other operating systems should not be considered an insurmountable task since Windows 2000 already includes logical drivers for I_2O.

[4] For example, Windows, LINUX, or a UNIX variant executing on Intel, SUN, RS/6000 or other processor platform.

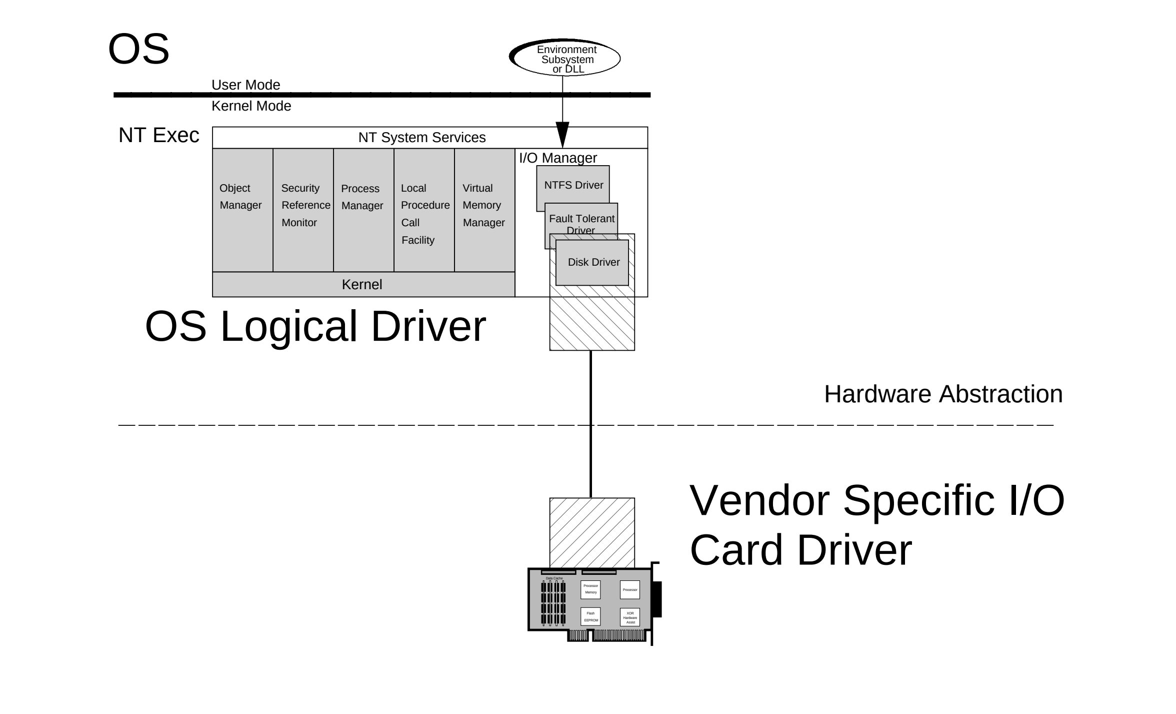

Figure 3. Driver Isolation

The key to this model is that logical I/O requests are managed by the operating system and that the physical I/O requests are managed by third party driver (right-hatched at the bottom of the figure) that is executed within the I/O card. Hence, should a driver within an I/O card experience an execution exception, the failure can not corrupt the operating system. Moreover, the resources required to perform these functions are provided by the I/O card.

The standard protocol engines introduced in the discussion of the IBA logical model are called host and target channel adapters. A channel adapter terminates a link and executes transport-level functions. The host channel adapter (HCA) supports the hardware communication interface that is managed by the lower level of the operating system's layered driver model. The HCA transmits logical I/O requests that are passed across the hardware boundary through the switched fabric to a target channel adapter (TCA) for execution. Figure 4 provides an overview of this relationship.

The target channel adapter (TCA) manages the I/O requests that are ***passed across the hardware boundary*** to it from the HCA via the switch. Depending on the total I/O bandwidth required, multiple I/O cards (e.g., fibre channel, SCSI, or Ethernet) may be incorporated in an I/O subsystem chassis. In an enterprise class implementation, multiple servers could ***share*** a set of TCAs (i.e., an I/O chassis) through a fabric switched network.

A collection of HCAs, TCAs, and switches are referred to as a subnet by the IBA. A subnet instance (which can define a network approximately 300 meters in diameter) may be comprised of 64K nodes[5] and multiple geographically dispersed subnets may be connected using routers. The nodes are connected using physical links that are rated at 2.5, 10, and 30 Gbit/Sec. These interconnections are referred to as 1X, 4X, and 12X links respectively.

While this brief overview has only laid out a few of the basic concepts of the InfiniBand architecture, *it is clear that the IBA provides a clean sheet solution to the server-scaling problem as well as a wide variety of other contemporary architectural issues.*

[5] An overloaded term used to refer to channel adapters, switches, or routers.

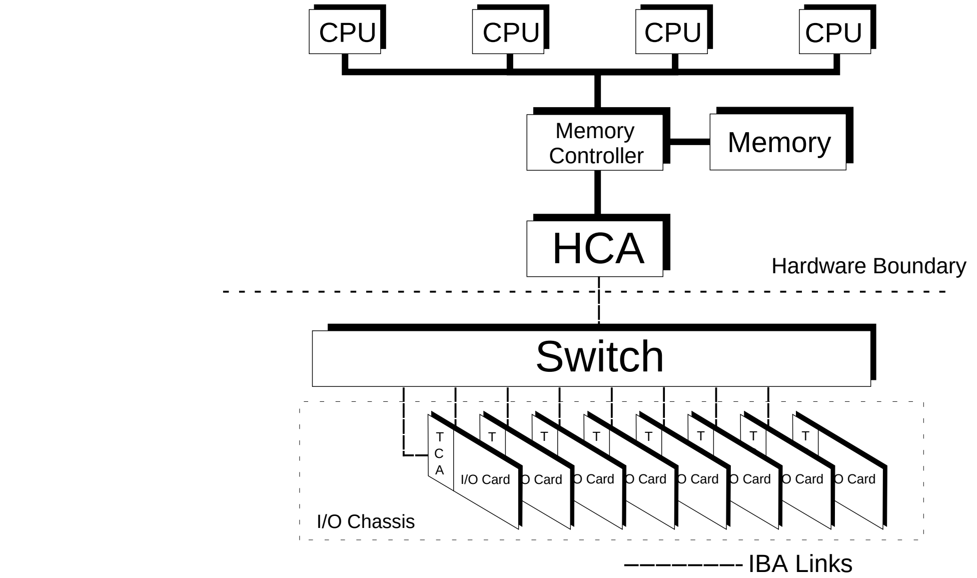

Figure 4. Host Channel Adapter / Target Channel Adapter

4. Hardware Measurement Primitives

As was discussed in the prior section, a subnet is a collection of nodes (i.e., HCAs, TCAs, switches, and routers) and routers may be employed to interconnect subnets. The IBA General Services Interface provides a variety of services for the management of subnets. These services include subnet administration, connection management, SNMP tunneling, baseboard management, device management, vendor specific services, and performance management.

By definition, every node must include a performance management agent that maintains counters as well as providing the means for sampling specific quantities over specified intervals. At a minimum, each node must maintain counters of erroneous and discarded packets as well as the number of subnet management packets that were dropped due to resource limitations. In addition, vendors may elect to include optional counters to differentiate and increase the value of their products. Commonly discussed optional counters include bytes and packets both transmitted and received.

Since the sampling mechanisms may be programmatically controlled through the general services interface, they provide a much greater potential wealth of measurement data. A generic sampling mechanism is shown in Figure 5. At a minimum, each node must provide one sampling mechanism with one counter. As a maximum, each node can support 256 sampling mechanisms, each of which can support 15 counters. Mechanisms are provided for determining the capabilities of a node, assigning it sampling quantities as well as duration, and then harvesting the counters at the end of the sampling interval.

It is important to note that the objective of the IBTA is to define an architecture, not specify the features and functions of the products, which will be created to exploit it. Hence, it may be difficult for some to envision how these measurement primitives can be employed to build a comprehensive measurement scheme for future servers.

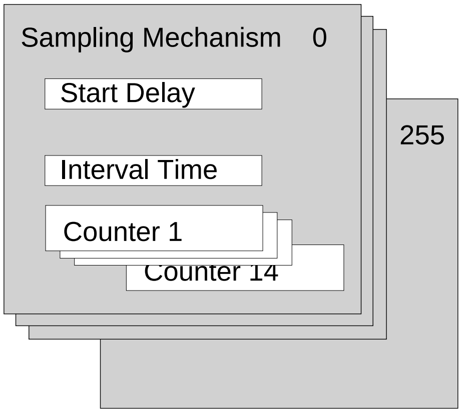

Figure 5. Node Sampling Mechanisms

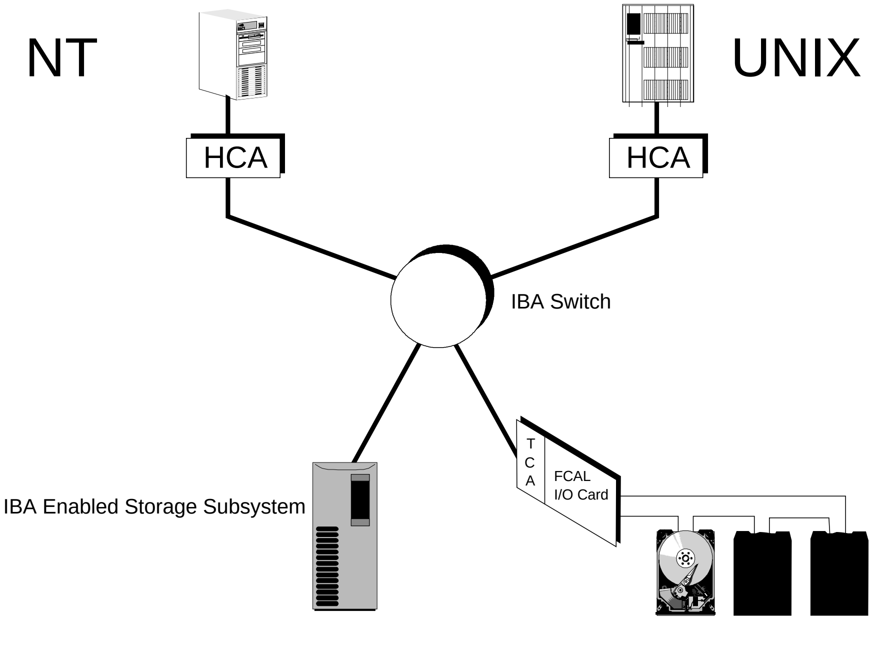

Figure 6. Hypothetical IBA Measurement Environment

Essentially, each node processes and decodes packets passed to it over the network. Hence, each node can be thought of as being an embedded protocol/activity analyzer. That is, the node is capable of counting and timing activity on the link as well as the characteristics of the packets it processes. Hence, if all of the nodes in a subnet are measuring in concert, a complete picture of the subnet's activity can be developed.

Figure 6 provides of a hypothetical IBA measurement environment. This simple environment is comprised of two servers (NT and UNIX), an IBA switch, a fibre channel arbitrated loop I/O card, and three FC SCSI drives. As a first observation, it is important to note that measurement is an IBA network general services application and is not dependent on the operating systems employed by the servers attached to HCAs. Hence, exactly the same hardware measurements would be available for each of the servers. Potential HCA measurements include HCA utilization as well as traffic (send/receive packets and bytes), data transfer time, and response by target TCA address.

At the switch, the traffic and utilization of each port could be measured. Finally, traffic and utilization measurements could be collected at the TCA. In addition, a storage TCA could collect back-end measurements of the service time components and utilization for the devices it manages.

While some may view this hypothetical environment as optimistic, it is important to note that system management is one of the foremost challenges for enterprise class server implementations. Hence, it is not an unrealistic expectation that vendors competing for market share in enterprise class environments will elect to differentiate their product offerings by adding extensive measurement and vendor specific facilities to their product offerings.

5. Comments

The InfiniBand Architecture represents an industry wide solution to a set of problems that present severe limitations to the future scalability of today's open system architectures. Moreover, it is a unified architecture for all vendors rather than a vendor specific solution that you are expected to adapt to the other resources in your environment. One key element of the IBTA is that their governance model requires that every member agree to a standard cross licensing agreement for all of the hardware and software elements developed by any of the IBTA members. The author hopes that this will herald the end of dead-end, proprietary, and vendor-specific solutions to enterprise wide problems.

From the perspective of measurement for open systems, the author elects to paraphrase Henry Kissinger's statement about the future of the Vietnam War in the fall of 1971.***I see the light at the end of the measurement tunnel!***

References

[1] IBA Specification Version 1.0, www.infinibandta.org .

[2] Shanley, T. and Anderson, D., ***PCI Systems Architecture***, Addison-Wesley, 1995.

[3] ***Technology Brief: PCI-X Technology***, Compaq Computer Corporation, February 1999

[4] **I_2O Special Interest Group**, www.i1osig.org.