Server I/O Networks Past, Present, and Future

Renato John Recio
Chief Architect, IBM eServer I/O
IBM Systems Group, Austin, Texas
recio@us.ibm.com

Abstract

Enterprise and technical customers place a diverse set of requirements on server I/O networks. In the past, no single network type has been able to satisfy all of these requirements. As a result several fabric types evolved and several interconnects emerged to satisfy a subset of the requirements. Recently several technologies have emerged that enable a single interconnect to be used as more than one fabric type. This paper will describe the requirements customers place on server I/O networks; the various fabric types and interconnects that have been used to satisfy those requirements; the technologies that are enabling network convergence; and how these new technologies are being deployed on various network families.

General Terms

Design, Experimentation, Performance, Standardization.

Keywords

10 GigE, Cluster, Cluster Networks, Gigabit Ethernet, InfiniBand, I/O Expansion Network, IOEN, iONIC, iSER, iSCSI, LAN, PCI, PCI Express, RDMA, RNIC, SAN, Socket Extensions, and TOE.

1. INTRODUCTION TO SERVER I/O NETWORKS

Server I/O networks are used by servers to connect to I/O devices, clients, and other servers. Servers use a wide range of interconnects to attach I/O, from traditional buses (e.g. PCI) to more complex general purpose networks that incorporate serial links, packet-routing, and switches.

2. SERVER I/O REQUIREMENTS

The requirements placed on Server I/O networks are derived from server requirements and include the following:

- Performance related requirements
  - Low latency - the total time required to transfer the first bit of a message from the local application to the remote application, including the transit times spent in intermediate switches and I/O adapters.
  - High throughput - the number of small block transactions performed per second. The size and rate of small block I/O depends on the workload and fabric type. A first level approximation of the I/O block size and I/O throughput rates required for various I/O workloads can be found in "I/O Workload Characteristics of Modern Servers" [4].
  - High bandwidth - the number of bytes per second supported by the network. Typically, used to gauge performance for large block data transfers. Peek bandwidth describes the bandwidth provided by a given link; whereas Sustained bandwidth describes the actual bandwidths provided after taking application, operating system, I/O adapter, and protocol inefficiencies into account. Again, a first level approximation of the bandwidth required for various I/O workloads can be found in "I/O Workload Characteristics of Modern Servers" [4] and ASCI Purple Statement of Work [1].
  - Host Overhead (a.k.a. efficiency, utilization) - a measure of the number of cycles expended by the host CPU to perform an I/O operation.
  - Memory Overhead (a.k.a. efficiency, utilization) - a measure of the number of times data must be moved through host memory, before it reaches the final consumer of the data.
- Connectivity related requirements
  - Scalability - ability to interconnect from a small number to a large number of endpoints.
  - Distance - ability to interconnect endpoints that span a wide range (from short to long).
  - Performance Density - ability to package a larger number of computing resources in a limited space, and is typically measured in performance/ft^3(/watt).
- Self-management related requirements
  - Self-healing (unscheduled outage protection) - ability to remain available despite the presence of failures. Unscheduled outages can be caused by transient (short duration) or permanent (long duration) faults. For transient faults, server I/O interconnects typically use a combination of data redundancy (e.g. cyclical redundancy checks) and signal quality inspection to detect the fault and use operation retries to recover from the fault. If a fault cannot be recovered through temporal recovery, it is typically considered a permanent fault. Server I/O interconnects typically recover from permanent faults through the use of redundant paths and path switchover that reroute traffic through faultless paths.
  - Self-configuring (scheduled outage protection) - ability to remain available during the addition, modification, or deletion of server I/O interconnects resources (e.g. links, switches, bridges, and other endpoints).

SIGCOMM'03, August 25-29, 2003, Karlsruhe, Germany. Copyright 2003

ACM 1-58113-735-4/03/0008...$5.00.

- Self-optimizing (Service Level Agreement, Quality of Service) - ability to provide a sustained level of service despite increases in fabric or resource utilization. For server I/O interconnects, several key self-optimizing requirements are:

  Fabric congestion management - static or dynamic mechanisms used to manage fabric utilization peaks.

  Service differentiation management - mechanisms used to match the performance levels associated with a given service to the performance levels provided by the end point and fabric.

- Self-protecting (a.k.a. secure) - ability to protect against unauthorized access.

- Cost related requirements

  - Infrastructure build-up cost - amount of infrastructure change required to support a given interconnect. Includes development effort (costs) and customer effort (e.g. new management tools).

  - Standardization - interconnect protocols and interfaces defined by a specification that is widely accepted and adhered to within the industry. Allows procurement of products from several (to many) different vendors.

  - Total Cost of Ownership - Hardware, software, and management costs associated with a given I/O interconnect. Standards interacts with volumes to lower TCO.

  - Fabric consolidation - ability to solve the needs of more than one fabric type, which reduces total cost of ownership. For example, fabric consolidation reduces the number of software management products and skills used to manage the system.

- Virtualization related requirements

  - Host virtualization - a mechanism that allows a single physical host to run multiple, independent operating system images (a.k.a. partitions) concurrently and provides access protection between the operating system images.

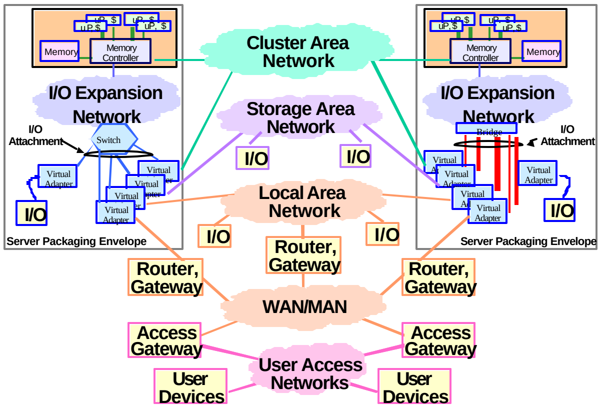
  - Interconnect virtualization - a mechanism that allows a single physical interconnect to be segmented into multiple, virtual interconnects and provides access protection between the virtual interconnects.

  - I/O virtualization - a mechanism that enables physical I/O unit's resources to be aggregated and managed in shared resource pools. Each shared resource pool is associated to one or more host. Only hosts that are associated with a shared resource pool are allowed to access the shared resource pool.

3. SERVER I/O FABRICS AND INTERCONNECTS

Server I/O network requirements conflict to some degree. Fully satisfying all the requirements with a single network has not been possible in the past, so several "**Fabric Types**" (on page2) and "**Interconnect Families**" (on page3) have proliferated. Figure1 shows a typical large server topology and the various Server I/O Fabric Types.

Figure 1.Server I/O Networks



3.1 Fabric Types

I/O Expansion Networks (IOENs) - These networks are used to connect the host processor complex to I/O adapters through the use of switches and/or bridges. IOENs allow large servers to attach a large number of I/O adapters, typically through a switched fabric. However, on a small server (esp. a server blade) the IOEN is typically a direct point-to-point interconnect between the memory controller and the I/O Bridge/Switch.

- Today, IOENs are proprietary and include: IBM's Remote I/O (RIO) network used on pSeries and iSeries servers[25]; IBM's Self-Timed Interface used on zSeries servers[12]; HP's Superdome I/O network[26]; HP's ServerNet[13]; and Silicon Graphics' XIO[23]. Some of these, for example IBM's RIO, are switched, link identifier based networks.

I/O Attachment - If the I/O adapter supports the link used by the I/O Expansion Network, then it can be attached directly to the IOEN. If the I/O adapter does not support the link used by the IOEN, then it is attached through an IOEN bridge

- Today, server I/O adapters primarily attach through a standard I/O bus (PCI family) and are used to either connect other fabric types or **directly attach storage** (e.g. disk, optical, tape) devices. Storage I/O devices today typically use parallel SCSI (Small Computer Systems Interface), parallel ATA (Advanced Technology Attachment), or Fibre Channel Arbitrated Loop (FC-AL). Though the industry is migrating the parallel links to serial (Serial Attached SCSI and Serial ATA).

Cluster Area Networks (CAN) support message-passing communications between servers in a cluster. Cluster Area Networks can be built from low-bandwidth, high-latency standard interconnects to high-bandwidth, low-latency proprietary interconnects. Examples of standard CAN interconnects include: Ethernet and, more recently, InfiniBand. Examples of proprietary CAN interconnects include: IBM's SP fabric [15]; IBM's InterSystem Channel; Myricom's Myrinet; Quadrics' QsNet; HP's ServerNet, and HP's Hyperfabric [14].

Storage Area Networks (SAN) connect servers to storage servers (a.k.a. storage subsystem). Storage servers attach storage devices through direct attached storage interconnects (i.e. SCSI, ATA, and FC-AL). Examples of SANs include: Fibre Channel [20], IBM's ESCON, and IBM's FICON.

Local Area Networks (LAN) connect servers to a wide variety of I/O devices, to other servers, and to clients, both locally or over the Internet. Though several standard LANs still exist today (Ethernet, Token Ring, FDDI, ATM), Ethernet has the highest volumes.

Metro & Wide Area Networks (MAN/WAN) and **Access Networks** connect servers to other distributed servers, and to clients, over long distances. **MAN/WAN** links typically have high bandwidths (133 Mb/s - 40 Gb/s) and span wide distances (10s of kilometers). **Access network** links typically are lower bandwidth (<50 Mb/s) and may include complex traffic conditioning and filtering mechanisms to handle the aggregation of access traffic onto MAN/WAN links. Examples of Access Networks, include: T1, T3, Cable, DSL, and fiber. Examples of MAN/WAN links include: Sonet, ATM, DWDM, and the emerging 10Gb Ethernet.

This paper will focus on IOENs, I/O Attachment, CANs, SANs, and LANs.

3.2 Interconnect Families

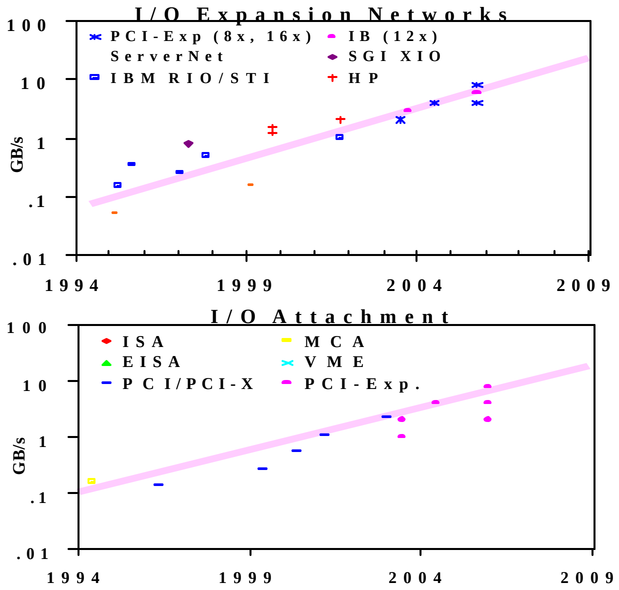
A single fabric type has not been able to satisfy the diversity of requirements. I/O is undergoing tremendous changes as technology enhancements and new technologies that solve a wider range of server I/O requirements emerge. Standards-based interconnects can enhance interoperability and lower cost, providing significant customer benefits. Advances in CMOS circuit density and performance have enabled industry-standard interconnects to expand their role. The advances are also enabling standard interconnects to satisfy a greater breadth of requirements simultaneously. The four major standards-based interconnects are:

- **PCI** (Peripheral Component Interconnect) Family is a low-cost interconnect that is used to attach memory mapped I/O adapters to desktop systems and servers.
- **FC** (Fibre Channel) is a switched-fabric interconnect that scales to satisfy the block-mode access needs of storage area networks.
- **Ethernet** is a low-cost, switched-fabric interconnect used in general purpose communications and storage networking. Several technologies are improving the performance characteristics of Ethernet: TCP/IP Offload Engines (TOEs), Remote Direct Memory Access (RDMA), low-latency switches, and higher bandwidth copper and fiber signaling (e.g. 10 Gb/s Ethernet). The combinations of these technologies provide low latency, zero copy message passing that approaches the performance levels required by SANs and CANs.
- **IB** (InfiniBand) is a switched-fabric interconnect that enables low-latency, zero-copy message passing for IOENs, CANs, and SANs.

4. IOENS AND I/O ATTACHMENT

As shown in figure 2, the bandwidth of I/O expansion networks has been increasing by 40% per year. Similarly, server I/O attachment links have been increasing by 40% per year for the past decade.

Figure 2.IOEN and I/O Attachment Performance



The primary factors enabling performance increases for both of these fabric types are: migrating to serial links, improving link rates, and increasing bus widths. The 40% per year increase is expected to continue for the next 5 to 7 years, at which point going beyond 10 Gigabits/second per line in copper will likely require architectural changes in the physical layer and migration to fiber.

The two primary interconnect families vying for acceptance in the IOEN and I/O Attachment markets are the PCI family and InfiniBand. The following sections will describe and compare these two interconnect families.

4.1 PCI family overview

The PCI Family offers a **Memory Mapped I/O (MMIO)** bus created to satisfy the I/O adapter needs of desktop and server systems. The high volume of the desktop market has enabled the PCI family to achieve the greatest number of supported adapters. The PCI family will likely remain the dominant architecture for I/O attachment.

PCI was originally defined as a parallel bus (32 or 64 bit width) with a single clock running at 33 MHz (and later 66 MHz).

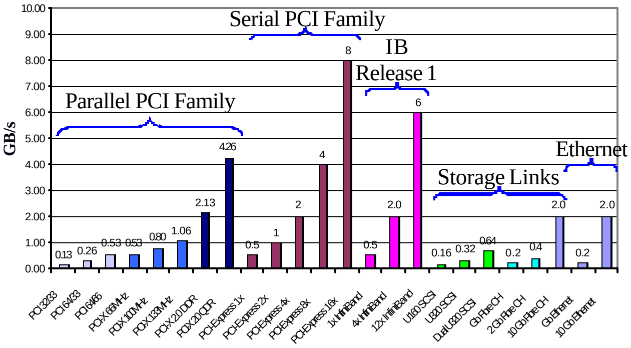
The PCI-X specification [28], released in September 1999, reduces overhead, increases frequency (to 133 MHz), and maintains electrical and software programming compatibility with PCI. PCI-X 2.0 will provide both electrical and software programming compatibility and will increase Parallel PCI bandwidth to 2.1 to 4.2GByte/s. PCI-X is designed as an I/O attachment link and is not suitable as a general-purpose I/O expansion network because it provides only chip-to-chip and card-to-card connections.

Increases in bus width and frequency have enabled the performance growth of PCI and PCI-X. The frequencies of current PCI links make maintaining a parallel-bus architecture increasingly difficult. Additionally, parallel PCI does not drive high bandwidth per pin, requiring 32-64 data pins to achieve high bandwidths.

Using a new, self-timed, serial link to provide scalability allows PCI Express [29] to avoid these limitations, providing up to 4Gbyte/s for a 16x link while using fewer pins. For MMIO adapters, PCI-Express provides software compatibility with PCI. In addition, PCI Express provides new capabilities that allow its use as an I/O expansion network, especially on low-end servers where higher end functions like redundant paths may not be as important (more on this in Section4.3).

As shown on Figure3 PCI-X and PCI-Express both scale to meet the peak bandwidth needs of the standard CAN, LAN, and SAN interconnects. Unfortunately, both suffer from performance bottlenecks because of their memory-mapped I/O architecture, which requires processor-synchronous programmed I/O (PIO) reads that traverse the network. A processor thread may stall until the PIO read operation completes, limiting its useful bandwidth. The performance impact is worse for larger configurations.

Figure 3.I/O Attachment Performance



PCI-Express allows I/O adapters to be packaged in a separate unit than the host processors, whereas PCI-X only provides chip-chip and card-card interconnection. Both interconnects have similar management attributes, though PCI-Express enables service differentiation through 8 traffic classes and virtual channels. PCI chips require a relatively small delta to support PCI-X, whereas PCI-Express represents a new core to implement all the various new layers (i.e. transaction, data link, and physical).

PCI-Express can be used as an IOEN, specially on lower end servers where redundant IOEN paths may not be as important.

For PCI-X and PCI-Express the host is responsible for performing host virtualization. This is typically accomplished either by: dedicating the adapter to a single host virtual server; or sharing the adapter through proprietary techniques, because neither standard defines standard I/O virtualization mechanisms. Efficient sharing of expensive I/O adapters is a key requirement for high-end servers. Achieving high performance on shared adapters may differentiate I/O products in the high-end server market.

As an I/O Attachment link, the migration from PCI-X 1.0 to 2.0 offers the path of least resistance for hardware vendors, This path will likely be used on 2004-2006 class servers. PCI Express will likely appear in desktop systems first and be used as a replacement for the Advanced Graphics Port and as an ASIC interconnect. PCI-Express will likely start to make in-roads into the server market in the late 2005, early 2006 time-frame.

Figure4 summarizes the attributes of PCI-X and PCI-Express as I/O Attachment links.

Figure 4.I/O Attachment Link Comparison

	PCI-X (1.0, 2.0)	PCI-Express
Performance		
Effective link widths	Parallel 32 bit, 64 bit	Serial 1x, 4x, 8x, 16x
Effective link frequency	33, 66, 100, 133, 266, 533 MHz	2.5 GHz
Bandwidth range	132 MB/s to 4.17 GB/s	250 MB/s to 4 GB/s
Latency	PIO based synchronous operations (network traversal for PIO Reads)	PIO based synchronous operations (network traversal for PIO Reads)
Connectivity		
Scalability	Multi-drop bus or point-point	Memory mapped switched fabric
Distance	Chip-chip, card-card connector	Chip-chip, card-card connector, cable
Self-management		
Unscheduled outage protection	Interface checks, Parity, ECC No redundant paths	Interface checks, CRC No redundant paths
Schedule outage protection	Hot-plug and dynamic discovery	Hot-plug and dynamic discovery
Service level agreement	N/A	Traffic classes, virtual channels
Virtualization		
Host virtualization	Performed by host	Performed by host
Network virtualization	None	None
I/O virtualization	No standard mechanism	No standard mechanism
Cost		
Infrastructure build up	Delta to existing PCI chips	New chip core (macro)
Fabric consolidation potential	None	IOEN and I/O Attachment

4.2 InfiniBand overview

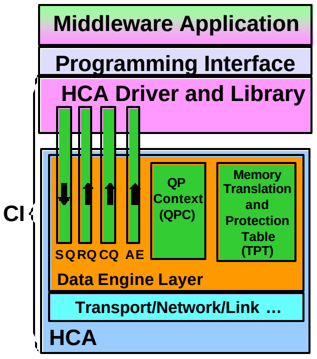
The purpose of the InfiniBand (IB) standard was to provide enterprise class network capabilities for high-end servers, such as virtualization, high availability, high bandwidth, enhanced manageability, and sharing. InfiniBand is a switched-fabric interconnect that enables low-latency, zero-copy message passing. The original fabric types targetted by IB were the Cluster Area Networks and an I/O Expansion Networks. Shortly after its creation, IB also targeted storage networks by using the SCSI RDMA Protocol (SRP), though SRP has not been aggressively embraced by the major storage vendors.

The major benefit of InfiniBand is its standardization of the following performance innovations:

- A memory and queue management interface that enables a zero-copy, zero-kernel interface to user space applications that reduces microprocessor overhead significantly;
- Using many communication pipelines (queue pairs) that directly access a fully offloaded network stack;
- Using message based asynchronous operations that do not cause processor thread stalls; and
- Using a serial, switched fabric that scales from 1x (250 MB/s) to 4x (1 GB/s) to 12x (3 GB/s).

Figure5 provides an abstract, overview of an InfiniBand Host Channel Adapter (HCA). A consumer (e.g. middleware application) running on the host interacts with the HCA software through OS programming interfaces. A portion of the HCA software may be executed in user mode and can directly access HCA work queue pairs and completion queues. A portion of the HCA software must be executed in privileged mode. The portion that executes in privileged mode includes the functions associated with QP and memory management. The QP management functions are used to create QPs and modify QP state, including the access controls associated with a QP. The memory management functions are used to register and modify the virtual to physical address translations, and access controls, associated with a memory region. The IB access controls enable user mode applications to directly access the HCA without having to go through privileged mode code. The Software Transport Interface and Verbs chapters of the IB specification define the semantics for interfacing with the HCA. The IB specification does not define the programming interfaces used to interact with the HCA. However, the Open Group's Interconnect Software Consortium is defining extensions to the Sockets API and a new Interconnect Transport API [19]. The latter provides more direct access to the IB HCA. The InfiniBand Architecture Specification Volume 1, Release 1.1 contains a more detailed description of the IB HCA [17].

Figure 5.IB HCA RDMA Service Overview



□SQ – Send Queue
□RQ – Receive Queue
□QP – Queue Pair
□QP = SQ + RQ
□CQ = Completion Queue

- Verb consumer – Software that uses HCA to communicate to other nodes.
- Communication is thru verbs, which:
  - Manage connection state.
  - Manage memory and queue access.
  - Submit work to HCA.
  - Retrieve work and events from HCA.
- Channel Interface (CI) performs work on behalf of the consumer.
  - Consists of software, firmware, and hardware.
  - Performs queue and memory mgt.
  - Converts Work Requests (WRs) to Work Queue Elements (WQEs).
  - Supports the standard IB layers (Transport, Network, Link, and Phy).
  - Converts Completion Queue Elements (CQEs) to Work Completions (WCs).
  - Generates asynchronous events.

To support 2006/07 time-frame servers, the line bandwidth will need to be upgraded. The upgrade will likely be to 5 or 10 Gb/s. Whether the upgrade is copper or fiber, will mostly depend on the ability of driving 6-10 meters over copper and the cost differential between copper and fiber.

IB also provides standard support for high-end enterprise server functions, such as:

- Hardware support for host, fabric, and I/O virtualization;
- Self-management mechanisms, such as:
  - Unscheduled outage protection through the use of interface checks, CRC, and port-level switchover; and
  - Scheduled outage protection through hot-plug, dynamic discovery, and host I/O assignment;
  - Service levels to support service differentiation;
- A network stack that can scale from a single server network to one that attaches many endpoints share; and
- System to I/O package isolation with point-to-point distances from 15 meters to kilometers via optical fiber.

However, IB requires a new infrastructure to be developed, deployed, and managed.

4.3 IOEN Comparison IB vs PCI-Express

PCI Express does not provide native mechanisms for virtualization. Sharing PCI-Express adapters between hosts either requires: additional processor cycles to provide the necessary access controls between host images, or a proprietary implementation. Whether vendors will continue using proprietary virtualization mechanisms for attachments that use memory-mapped I/O, such as PCI-X and PCI Express, is not clear. InfiniBand provides native virtualization mechanisms that do not consume additional processor cycles.

The latency from synchronous I/O operations are another disadvantage of the PCI family's using memory-mapped I/O. Server vendors have eliminated most of the latency from PIO writes by posting them to hardware near the host and completing them later. Still, server and adapter vendors have not been as successful at reducing the latency from with PIO reads. Using even a single PIO read per transaction can reduce performance significantly by stalling the processor for 100s of nanoseconds to microseconds, which reduces the effective bandwidth of the adapter. When InfiniBand is used as an I/O expansion network and for I/O attachment, its channel (Send and RDMA) model does not have this disadvantage because PIO operations never traverse the InfiniBand fabric.

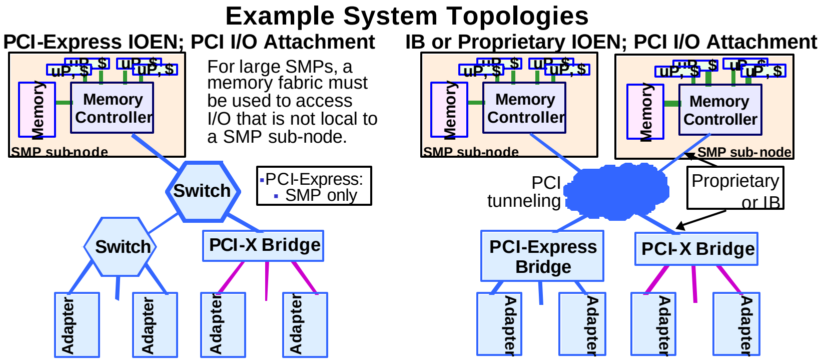
PCI-Express provides a root-tree based network topology, where all I/O attaches, through switches and bridges, to a root complex. The root complex contains one or more processors, and their associated memory subsystem. The PCI standard does not provide the mechanisms needed to support: multiple root complexes within a single SMP (a.k.a. SMP sub-node), or multiple root complexes between cluster nodes. Additionally, redundant paths are not defined in the standard. InfiniBand does not prescribe a specific network topology and supports redundant paths to I/O. For example, on IB multiple SMP sub-nodes can each provide redundant paths to the same I/O unit.

Although, InfiniBand offers significant performance and functional advantages for I/O attachment, it requires a new infrastructure in contrast to PCI Express, which uses the existing PCI software infrastructure. Still, the current definition of PCI Express neglects critical high-end server requirements, such as multipathing. Without these functions, as an I/O Expansion Network, PCI Express will be useful only on low-end servers. High-end servers will likely use proprietary links or InfiniBand with vendor-unique transport operations that enable tunneling of memory-mapped I/O operations. I/O Attachment will likely be through the PCI family (PCI-X, followed by PCI-Express), though some high-end I/O may attach directly to IB.

Figure6 compares IB to PCI-Express as an IOEN. It also compares the possible IOEN topologies for IB and PCI-Express. IB enables a multi-root I/O structure. As a result, in a large SMP configuration all I/O can be accessed by any sub-node connected to the IB fabric. In contrast, PCI-Express does not support a multi-root I/O structure. As a result, in a large SMP configuration some processors would have to channel non-local I/O through the memory fabric.

Figure 6.I/O Expansion Network Link and Topology Comparison

	PCI-Express	IB
Performance		
Effective link widths	Serial 1x, 4x, 8x, 16x	Serial 1x, 4x, 12x
Effective link frequency	2.5 GHz	2.5 GHz
Bandwidth range	250 MB/s to 4 GB/s	250 MB/s to 3 GB/s
Latency	PIO based synchronous operations (network traversal for PIO Reads)	Message based asynchronous operations (Send and RDMA)
Connectivity		
Scalability	Memory mapped switched fabric	Identifier based switched fabric
Distance	Chip-chip, card-card connector, cable	Chip-chip, card-card connector, cable
Self-management		
Unscheduled outage protection	Interface checks, CRC No redundant paths	Interface checks, CRC Redundant paths
Schedule outage protection	Hot-plug and dynamic discovery	Hot-plug and dynamic discovery
Service level agreement	Traffic classes, virtual channels	Service levels, virtual channels
Cost		
Infrastructure build up	New chip core (macro)	New infrastructure
Fabric consolidation potential	IOEN and I/O Attachment	IOEN and high-end I/O attachment
Virtualization		
Host virtualization	Performed by host	Standard mechanisms available
Network virtualization	None	End-point partitioning
I/O virtualization	No standard mechanism	Standard mechanisms available



For large SMPs, a memory fabric must be used to access I/O that is not local to a SMP sub-node.

Key PCI-Express IOEN value proposition

- Bandwidth scaling
- Short-distance remote I/O
- Proprietary based virtualization
- QoS (8 traffic classes, virtual channels)
- Low infrastructure build-up
  - Evolutionary compatibility with PCI

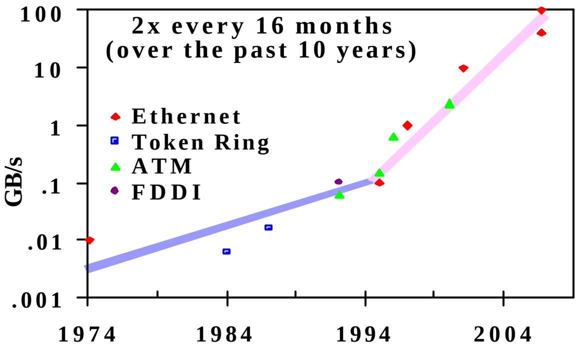
Key IB IOEN value proposition

- Bandwidth scaling
- Long distance remote I/O
- Native, standard based virtualization
- Multipathing for performance and HA
- QoS (16 service levels, virtual lanes)
- Message based, asynchronous operations
  - with many communication pipelines, and zero copy, user space I/O

5. LOCAL AREA NETWORKS

Ethernet has become the dominant standard for local area networks, eclipsing Token Ring, ATM, and FDDI. Figure7 shows the various LAN links that have been used over in the past. Improved signaling techniques, increased circuit density, and faster circuits have enabled LAN bandwidth growth of 60% per year. Optical links will enable this growth rate to continue even though copper links are reaching fundamental limits in frequency and distance. High-volume components and increasing circuit density will continue to improve networking price/performance.

Figure 7.LAN Performance Growth



Similarly, the Internet protocol (IP) suite has become dominant. The IP suite includes the transport, network, and management protocols. The transport protocols transfer data from an endpoint. The networking protocols route data between endpoints. The management protocols initialize, control, monitor, and maintain network components. The Internet Engineering Task Force (IETF) is the standards group for the IP suite. Although the IETF does not define specifications for IP suite offload, the IETF is creating a transport-layer function (remote direct data placement, a.k.a. remote direct memory access) that targets hardware-offload implementations. In addition, vendors have recently begun to offer network interfaces that offload IP suite functions.

IP networks include a wide range of components, which fall into two groups, endpoints and intermediate components. Network endpoints are typically servers, clients, or management components. The servers include general-purpose servers, such as zSeries servers or Intel-based servers, and special-purpose servers, such as file servers. Network intermediate components include switches, routers, and network appliances. Switches and routers transfer data through the network, and may also provide additional function, such as filtering. Network appliances provide additional functions, such as cryptographic functions.

6. ETHERNET OVERVIEW

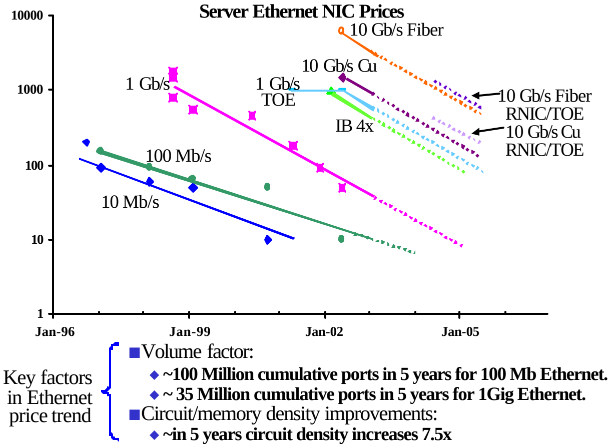
A host typically uses a Network Interface Controller (NIC, a.k.a. PCI Ethernet Adapter) to interface with Ethernet. Ethernet is a switched fabric interconnect that scales to satisfy the needs of Local Area Networks. Ethernet was originally defined, in 1975, as a self-timed, serial link running at 10 Mb/s over copper. Over the past decade, Ethernet bandwidth has increased to 100 Mb/s, 1 Gb/s, and more recently 10 Gb/s.

The key value proposition of Ethernet consists of:

- Low cost components from high volumes. Server volumes have enabled 1 Gb/s Ethernet NICs to drop to $50. Similarly in the near future, server volumes will fuel the reduction in 10 Gb/s Ethernet NIC prices (see Figure8).
- A very comprehensive support infrastructure, that is widely deployed for local area networks and is branching out into SAN and High Performance Cluster (HPC) Networks.
- Hardware supported host, fabric, and I/O virtualization.
- High availability through session, adapter, or port level switchover.
- Dynamic congestion management when combined with IP Transports.
- Long distance links (from card-card to 40 Km).
- High performance (when combined with TCP Offload).
- Scalable bandwidth (from 100 MB/s to 1 GB/s, and in the future, 4 GB/s).

Vendors are offloading TCP/IP from the host onto the NIC and using IB like mechanisms (with enhancements) to reduce significantly the processor and memory resources needed to perform storage, cluster, and LAN communications.

Figure 8.Ethernet NIC Prices



The next Ethernet generation (10GigE) will use 10gigabit/s links. 10GigE will satisfy a broad set of network markets, from WAN to LAN to bladed-server mid-planes.

The 10Gbit/s standard [16] defines one copper and four optical physical layer options (PHYs), supporting various link distances. The optical link PHYs defined for connecting 10GigE endpoints have been optimized to cover link distances from 300m (available today for roughly $300) to 40km (currently available for more than $1000). The prices of fiber transceivers will likely drop significantly (to the $150 to $200 range), as volumes ramp up on 10GigE and NIC vendors exploit higher density circuits.

The copper physical layer (XAUI, which uses four differential pairs at 3.125 Gbit/s each) is an on-card electrical link that is viable for in-the-box wiring. Several vendors, including Marvell [2] and Velio Communications [37], have produced XAUI transceivers that support a 15meter cable, which is actually the standard InfiniBand 8-pair, 24-gauge copper cable. A XAUI-based copper cable will likely be standardized within the next year. The transceivers for the copper version can be packaged on the same chip as the 10GigE media-access controller (MAC). Figure9 summarizes the attributes of 10GigE adapters.

The Ethernet community [10] is currently debating the next step in the Ethernet roadmap. The bandwidths of the top link candidates are 40Gigabit/s [24] [33] [38], which is well-established in the WAN industry, and 100Gigabit/s, which would follow the Ethernet trend of 10x per generation. The

Figure 9.Attributes of 10GigE adapters

10 GigE Copper Options

Phy Layer	XAUI
Intended use	Between cards.
Phy type	4x Copper Differential Pairs
Speed per link	3.125 Gb/s (8b/10b encoded)
Distance	57 cm
Transmitter	Copper
Transceiver Relative Price	0 – Embedded

10 GigE Fiber Options

Phy Layer	SR/SW	LX4/LW4	LR/LW	ER/EW
Intended use	Data Center and LAN	Data Center, LAN, and WAN	LAN and WAN	LAN and WAN
Phy type	1 Fiber 850 nm Wavelength	4 Fiber 1300 nm Wavelengths	1 Fiber 1300 nm Wavelength	1 Fiber 1550 nm Wavelength
Speed per link	10 Gb/s	3.125 Gb/s	10 Gb/s	10 Gb/s
Distance	300 m	300 to 10 Km	2-10 Km	40 Km
Transmitter	VCSEL	Laser	Laser	Laser
Transceiver Relative Price	1x ($100-200)	2x ($300-600)	2x ($200-600)	4x ($2000)

next step will likely be introduced four to five years after the servers begin shipping 10GigE.

Given the above considerations, Ethernet will clearly continue to dominate the LAN market. However, traditional Network Interface Controllers (NICs) do not provide the performance necessary for the 10GigE generation. As a result, the industry is focusing on IP suite offload.

6.1 Internet Protocol Suite Offload

The Internet Protocol (IP) suite comprises standards that define the protocols for exchanging messages and files over the Internet and local area networks. The Internet Engineering Task Force (IETF) defines IP standards, including transport, network routing, and management protocols.

Internet protocol processing for TCP/IP, UDP, iSCSI, and IPsec is consuming an increasingly large proportion of processor resources on many Web servers, multi-tier servers, network infrastructure servers, and networked storage servers.

The growing overhead from protocol processing has made changes to network interfaces necessary. Internet protocol processing for TCP/IP, UDP, iSCSI, and IPsec consumes an increasingly large portion of processor resources on many Web servers, multi-tier servers, network infrastructure servers, and networked storage servers. The portion of processor resources is increasing because the link speed has been growing faster than microprocessor performance and applications have become more network intensive. Memory latency, a critical parameter of protocol-processing performance, has remained nearly constant. An additional factor driving changes in network interfaces is increasing CMOS circuit densities, which now allow more functions to be integrated into adapter controller chips at a low cost.

Early network interfaces, such as the 10Mbit/s interface in 1994, were very simple, providing a controller, physical access function, a simple packet buffer, and a mechanism to interrupt the microprocessor. The host microprocessor performed the entire protocol stack, including the checksum and other error detection, flow control, segmentation and reassembly, data copying, and multiplexing packets across user applications.

Today, high-function network services are emerging that perform parts of the protocol stack:

- **Ethernet Link Service** Under this service the host performs TCP/IP processing, with some functions offloaded to the NIC, such as checksum processing, TCP segmentation, and interrupt coalescing. This service is available from most NIC vendors and is shipping on modern servers. This service does not completely eliminate receive-side copies. It usually has a proprietary programming interface, because this level of function has not been standardized.
- **TCP/IP Offload Engine (TOE) Service** This service typically performs all processing for normal TCP/IP data transfers. In addition, some of NICs that offer a TOE Service perform all processing for managing TCP/IP connections and errors. These NICs reduce host processor overhead and can eliminate receive-side copies in some cases. They typically require buffering to reassemble incoming packets that arrive out of order. TOE Service vendors include Alacritech, Lucent, Adaptec, QLogic, and Emulex. The distribution of functions between the host and the NIC, as well as the TOE Service interface, varies by vendor. The lack of standard semantics and interface for the TOE Service has hindered its adoption.
- **Remote Direct Memory Access (RDMA) Service** This service is accessed through an abstract RDMA verbs interface that defines memory and queue management semantics and associated access controls. The verbs do not specify the syntax of the RDMA Service interface. When implemented over the IP suite, after a connection is established, this service performs the following protocols: RDMA (Remote Direct Memory Access), DDP (Direct Data Placement), MPA (Markers with PDU Alignment), TCP, and IP. The RDMA Service requires less buffering than the TOE Service because it uses application buffers to reassemble packets that arrive out of order.

  The first generation of the RDMA, DDP, and MPA protocols has been standardized by the RDMA Consortium (RDMAC) and has been accepted as official work-group drafts by the IETF's Remote Direct Data Placement (RDDP) work-group. The RDMAC also standardized the interface for the RDMA Service, known as the verbs layer and has submitted the verbs to the IETF. A NIC that supports the RDMA Service is known as an **RDMA enabled NIC (RNIC)**. NICs that support the RDMA Service are under development by several vendors and are not available yet.

  The RDMA Service can only be used if both sides of a connection support it. Some Operating System (O/S) vendors will likely support the RDMA Service in their O/S code in order to enable a server that contains an RNIC to communicate with a client (or server) that doesn't.

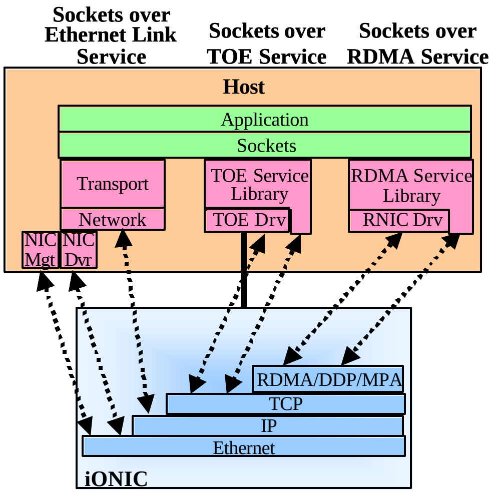
internet Offload NICs (iONICs, pronounced "i onyx") are NICs that support more than one of these services, and possibly other services, such as: IPSec, iSCSI, or iSER. Future server NICs will likely be **iONICs**.

Figure10 depicts the three services described above and shows where the functions are performed for each.

6.2 RDMA Service Comparison

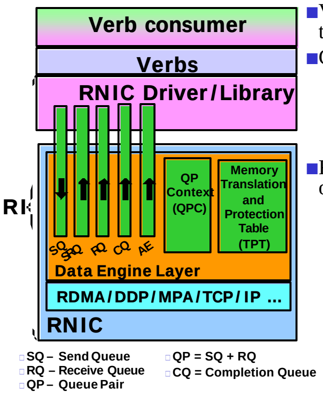
Though IB will obviously not be used in local area networking, it is useful to compare the IB's Reliable Connected RDMA

Figure 10.IP Offload Service Examples



Service [17] with the RDMA Service on an RNIC [32], or an iONIC, if more than one offload service is supported. As shown in Figure11, RNIC's have a similar verb interface [11] [31] as InfiniBand Host Channel Adapters (previously shown in Figure5). An RNIC uses TCP as the transport, whereas an HCA uses IB's Reliable Connected (RC) transport. The difference in the transports used between the two accounts for most of the operation ordering differences that exist between the RNIC verbs and the IB HCA verbs. The RNIC verbs did not incorporate all the functions defined in IB. For example, the RNIC verbs don't support: transports analogous to IB's Reliable Datagram and atomic operations. However, the RNIC verbs extended the functions provided by the IB verbs. For example, the RNIC verbs support: zero based offset memory registration; fast registration; memory region invalidation operations; shared receive queues; and work request lists. Some companies are pursuing the inclusion of these functions on a future IB specification release.

Figure 11.RNIC RDMA Service Overview



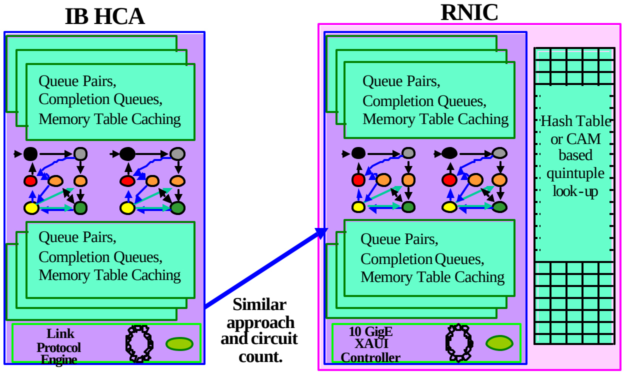
- Verb consumer – SW that uses RNIC to communicate to other nodes.
- Communication is thru verbs, which:
  - Manage connection state.
  - Manage memory and queue access.
  - Submit work to RNIC.
  - Retrieve work and events from RNIC.
- RNIC Interface (RI) performs work on behalf of the consumer.
  - Consists of software, firmware, and hardware.
  - Performs queue and memory mgt.
  - Converts Work Requests (WRs) to Work Queue Elements (WQEs).
  - Supports the standard RNIC layers (RDMA, DDP, MPA, TCP, IP, and Ethernet).
  - Converts Completion Queue Elements (CQEs) to Work Completions (WCs).
  - Generates asynchronous events.

The first generation of 10GigE RNICs will have function, performance, and cost that are equivalent to those of an InfiniBand 4x host channel adapter. The queue and memory management models for RNICs are a superset of the models for IB HCAs. However, IB's Reliable Connected transport service is simpler than the RNIC's RDMA transport, because the latter has to deal with: out of order reception, quintuple look-up, resegmenting middleboxes, IP packet fragmentation, and enhanced memory operations. As a result, an RNIC requires comparable logic as IB HCA, but, to deal with IP idiosyncrasies the RNIC requires additional state.

Figure12 compares the logic and memory differences between an IB HCA and an RNIC. The major difference in cost between an IB HCA and an RNIC depends on the link distance supported.

- For server-server or server-storage (e.g. iSCSI) communications within the data center, both the chip and card cost are the same for both. The RNIC would use a XAUI copper interface, which is similar to the IB 4x copper interface, to communicate between nodes. A copper (or fiber) cable would be used between racks.
- For internet and long-distance communications, the RNIC chip would cost the same, but the card would cost more. The additional cost comes from the eddy buffers and the likelihood that security protocols will also need to be offloaded. The eddy buffers are needed to handle: upstream link (e.g. PCI) congestion, LAN congestion, resegmenting middleboxes, and IP fragmentation.

Figure 12.Comparison of IB HCA and RNIC designs


IB HCA
RNIC
Queue Pairs, Completion Queues, Memory Table Caching
Queue Pairs, Completion Queues, Memory Table Caching
Queue Pairs, Completion Queues, Memory Table Caching
Queue Pairs, Completion Queues, Memory Table Caching
Hash Table or CAM based quintuple look-up
Link Protocol Engine
10 GigE XAUI Controller
Similar approach and circuit count.

- Both require about 71 M cells (including arrays used for context caching)
  - Requires 9x9 square millimeter die in 90nm technology
  - A 2006/07 product can be implemented in a 5x5 sq-mm die.
- Both require additional memory to store queue and memory management state
- RNIC requires additional memory (internal and external) to handle Quintuple look-ups and out of order reception.

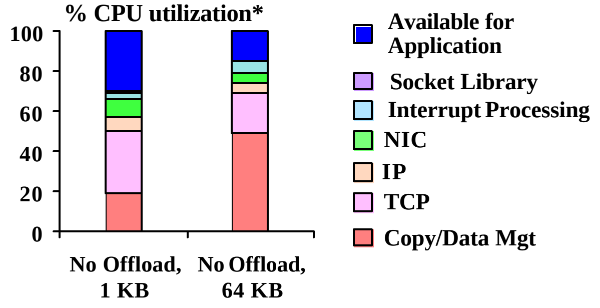
6.3 IP Offload Performance Benefits

Except where noted, the RDMA Service analysis performed in this section applies to both RNICs and IB HCAs.

Several published papers have described the resources (CPU and memory) expended in processing the TCP/IP stack [6] [7] [9] [21] [22]. These papers have also shown the efficiency gained by offloading TCP/IP and removing receive side copies [3] [5]. Within IBM, folks have made many similar measurements. Figure13 shows two such measurement for TCP Segment processing over a NIC that supports the Ethernet Link Service.

Each case depicted in Figure13 was measured on Linux running the Tuxedo web server with 512 clients. Since, for a web server, the server side measurements don't include large incoming sends, the copy overhead for each case was measured separately. The measurements found that for a 1 KB transfer using the Ethernet Link Service, the host spent: 40% of its CPU cycles performing TCP/IP processing; and 18% on copy and data manipulation. For a 64 KB transfer, the host spent: 25% of its CPU cycles performing TCP/IP processing; and

Figure 13.TCP/IP Overhead

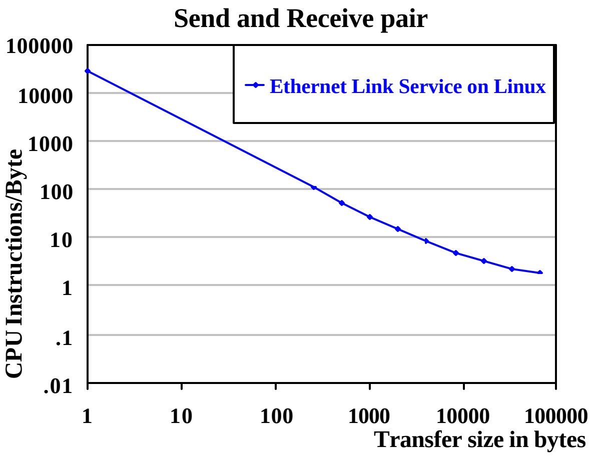


49% on copy and data manipulation. The incoming TCP/IP traffic traversed the host memory bus three times: once into a privileged buffer, once out of the privileged buffer, and once into the user space buffer. If the application needs the data to be contiguous, an additional data move through the host memory bus may be necessary to separate application headers from data.

Obviously, the application used to create the load shown in Figure13 is extremely network intensive. For applications that are much less network intensive, a larger percent of the CPU would be available for the application.

In the Ethernet Link Service measurement the host CPU performed an estimated 27,430 instructions for a 1 KB Send and Receive pair, resulting in 26.8 CPU instructions per byte transferred. As the transfer size increases the CPU instructions per byte decreases, because a portion of the network stack code path is traversed only once per segment or per frame. Figure14 shows this case.

Figure 14.Ethernet Link Service Overhead



Host network stack overheads can be significantly reduced through a combination of network stack offload and receive side copy removal:

- **Network stack offload**, through either a TOE or RDMA Service, moves the TCP/IP path length from the host to the network adapter. However, to gain the full benefit of network stack offload:
  - the adapter's network processing needs to be comparable (or faster) to the host's (e.g. by using hardware state machines to implement portions of the stack); and
  - the connection's lifetime needs to be sufficiently greater than the time it takes to: create a TOE or RDMA Service instance and transfer the connection from the host to the iONIC.
- **Receive Side Copy Removal**, through either a TOE or RDMA Service, eliminates the host processor cycles spent on copying incoming data from privileged to use buffers. Receive side copy removal is dependent on:
  - The application's Data Transfer Format.

    Single object transfer. Application transfers entire object using a single application Protocol Data Unit (PDU), which consists of one header and the data object.

    Segmented object transfer. Application segments the object into multiple application PDUs, where each PDU consists of one header plus an object segment.

    For the above two cases, several *PDU formats* are possible: 1) fixed header, fixed data size; 2) fixed header, fixed data size, except for the last data object, which has a variable length; 3) fixed header, variable data size; 4) variable header, variable data size; and 5) variable header, variable data size.
  - The application's Buffer Posting Model.

    Proactive Buffer Posting Model - The application always has at least one socket read outstanding.

    Reactive Buffer Posting Model - The application issues a Socket Read with Peek to look at a length field in the header and then posts one or more buffers to retrieve the header plus the length's worth of data.

    Inactive Buffer Posting Model - The application issues socket reads after the data has arrived.

    Of course, some application may use a mix of the above. For example, consider an application that doesn't have a buffer flow control mechanism. Such an application may typically use the Proactive Buffer Posting Model. However, a burst of incoming segments may cause relatively short inactivity in the application's buffer posting code.
  - Operating System socket enhancements.

    Asynchronous I/O API - For Socket Reads, the Operating System (O/S) posts the read operation and returns control to the application. A call back (or poll mechanism) is used to determine completion of the read operation.

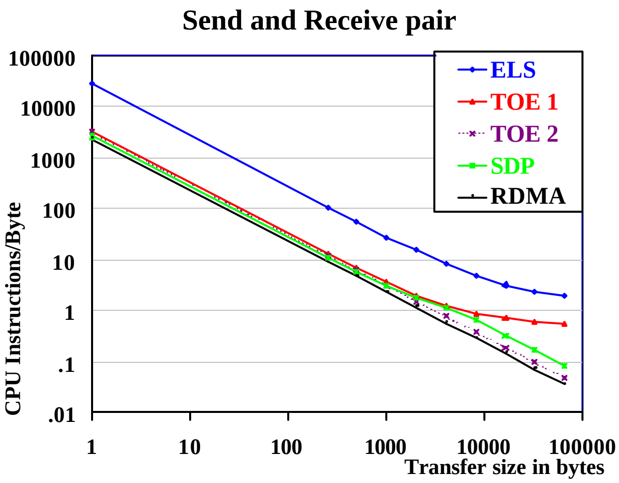
    Sockets Direct Protocol (SDP) - A protocol used to convert the socket stream interface into a flow controlled, message based protocol that performs implicit memory registration and uses Send/Receive and RDMA operations that are available under the RDMA Service.

    Explicit Memory Registration API - Exposes the memory registration mechanisms available under the RDMA (and possibly TOE) Service.

    Explicit RDMA Service Access API - Exposes the RDMA Service's Send/Receive, RDMA Write, and RDMA Read operations to the application.

Though a full paper describing all of the combinations can be written, Figure15 summarizes the analytical modeling results for some key combinations of the above improvements:

Figure 15. TOE and RDMA Service Overheads



- ELS - Same Ethernet Link Service case as Figure14 (the application models and O/S socket API enhancements have a relatively small impact on efficiency).
- TOE 1 (TOE Service 1) - Copies are not removed. Under this case:
  - The application uses any of the data formats described in this section. It also uses the Reactive or Inactive Buffer Posting Model over a TOE Service.
  - The O/S supports an Asynchronous I/O, socket API. The O/S and TOE Service provide implicit or explicit memory registration (relative small impact between the two).
- TOE 2 (TOE Service 2) - Copies are removed. Under this case:
  - The application uses the Proactive Buffer Posting Model over a TOE Service. The receiving application either can predict the incoming PDU format or does not need to receive the object in a contiguous (virtual address) range.
  - The O/S sockets implementation supports the Explicit Memory Registration API. The O/S and TOE Service provide explicit memory registration.
- SDP (over an RDMA Service) - Copies are needed for small data transfers. Under this case:
  - The application uses a Proactive or Reactive Buffer Posting Model. The receiving application uses any of the data formats described in this section.
  - The O/S's sockets implementation supports an Asynchronous I/O API and SDP over an RDMA Service. RDMA Reads and RDMA Writes are used to remove copies for larger transfers.
- Socket RDMA - Copies are fully removed. Under this case:
  - The application uses any of the data formats described in this section and any of the three buffer posting models.
  - The O/S's socket implementation supports the Asynchronous I/O, Explicit Memory Registration, and Explicit RDMA Service Access socket extensions. The RDMA Service provides RDMA Reads and RDMA Writes.

The TOE 1, TOE 2, and SDP cases do not require application modifications, where as the Socket RDMA case does require application modifications.

As summarized in Figure15, for long lived connections, a TOE or RDMA Service can remove over 90% of the instructions executed by the host's network stack per data transfer. Additionally, for an Ethernet Link Service that fully utilizes the link, the host memory to link bandwidth ratio is 3. That is, the host memory bandwidth needs to be 3 times greater than the link bandwidth. Whereas the TOE and RDMA Services can remove the need for a copy and reduce the memory to link bandwidth ratio down to 1. However, for the TOE Service, and to a far less extent SDP, the mileage varies based on the application's data format and buffer posting models. In contrast, the full benefit can be realized if the RDMA semantics are exposed to the application, either through the Socket protocol or a native API (such as the Interconnect Software Consortium's Interconnect Transport API). Of course the application would have to change to support explicit RDMA semantics, some already have (e.g. databases), others probably will not for quite some time, maybe ever (e.g. TelNet).

6.4 LAN Summary

Standardization of hardware offload mechanisms is well underway in the IETF. Ethernet adapter vendors have already begun shipping NICs with a TOE Service and several vendors are now developing an RDMA Service. Currently, the RDMA and TOE Services can be embedded with a 10 GigE controller on a 9x9 sq-mm chip. The small chip size enables server blade vendors to use a copper 10 GigE backplane to interconnect blades and, through emerging 10 GigE cable standards, drawers. As the RDMA and TOE Services mature, some vendors will embed IP offload into the memory subsystem and eventually the CPU.

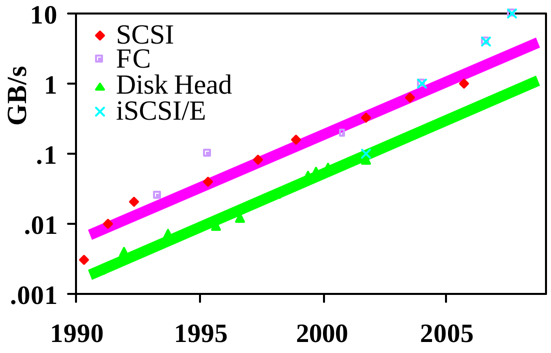
To fully exploit the performance possible with these offload services, operating system vendors will enhance their socket interfaces and, overtime, applications will make the modifications necessary to fully optimize performance.

7. STORAGE NETWORKS

The majority of storage devices (disks, tapes, optical) continue to be attached through parallel interfaces (SCSI and IDE). Servers use internal memory mapped I/O adapters or external remote controllers to attach these storage devices. The internal adapters and external controllers attach the devices directly or through a network. The migration from direct-attached storage to networked storage is well underway. Several storage networks are currently used on enterprise servers, including Fibre Channel, FICON, ESCON, and others. More recently IP/Ethernet has been used to network storage. Figure16 depicts performance growth for storage links over the past 12 years.

Ethernet offers a powerful value proposition to enterprise storage networking. Because enterprises typically have an Ethernet management infrastructure in place, using that infrastructure for storage can reduce the number of networks that need to be managed. Ethernet supports more advanced functions, such as differentiated services, security, and management, than many alternatives, including SCSI and Fibre Channel. However, Ethernet faces two major inhibitors when compared to FC: a higher host CPU overhead that is associated with performing iSCSI and TCP/IP processing; and higher switch latencies. This section will explore these two inhibitors in more detail.

Figure 16. Storage link bandwidth growth



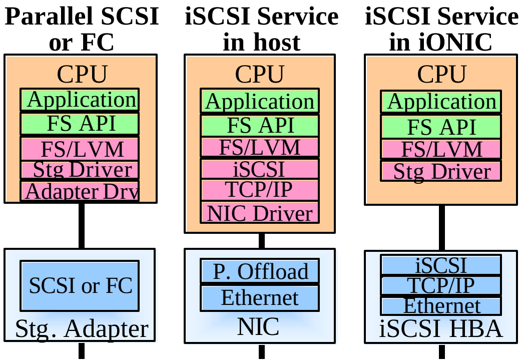
7.1 Storage access models

Servers currently use either block-mode I/O or file-mode I/O to access storage. With block-mode I/O, the storage device presents the server with a linearly addressed, fixed record size, and the server performs I/O operations that access one or more of these records. With file-mode I/O, the storage device provides the server with a higher level mechanism for creating, managing, modifying, and destroying directories and variable-size files. Interconnects that support block-mode I/O include parallel SCSI and Fibre Channel. LAN interconnects, such as Ethernet, have been the primary interconnects for supporting file-mode I/O. More recently, LAN interconnects are also supporting block-mode I/O.

7.1.1Block Mode access over Ethernet

Network interfaces that offload IP processing will be required for block-mode storage access over Ethernet because adding Internet processing to the storage overhead would increase the processor utilization on hosts and storage servers. Figure17 depicts two of a few approaches that can be taken to distribute the functions required to support block mode storage over IP/Ethernet.

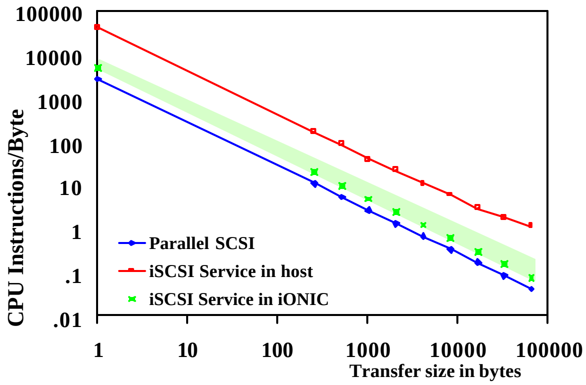
Figure 17. Several block mode storage access options



- *iSCSI (iSER) Service in host (over Ethernet Link Service in NIC).* This approach uses the microprocessor in the host (initiator) or storage subsystem (target) to perform all the iSCSI and TCP/IP processing. Early iSCSI storage products took this approach, because it provided fast time to market.
- *iSCSI (iSER) Service Offloaded to iONIC.* Under this approach the iSCSI (or iSER) and TCP/IP processing are offloaded to the adapter. The adapter implementation uses a combination of firmware and hardware to implement iSCSI (iSER). To satisfy performance requirements, the data placement mechanism needs to be implemented in either: a hardware state machine (vs. performed in firmware) or a very high performance microprocessor. For iSCSI, the adapter uses a data placement mechanism that is unique to iSCSI. For iSER, the adapter uses an RDMA based data placement mechanism. Some adapter vendors may provide optimal performance for both mechanisms, others may optimize performance for one of these mechanisms. For example, a vendor may implement the iSCSI data placement mechanism in a state machine and the iSER data placement mechanism using a slightly slower microprocessor.

Though I am not at liberty to provide measurements for the three cases in Figure18, I am able to provide code path estimates that are based on several storage device drivers I have written prior to working on I/O architecture. Figure18 summarizes the analytical modeling results based on my code path estimates.

Figure 18. Block mode I/O overhead



- Parallel SCSI - The host (SCSI initiator) path length for a SCSI device driver transaction (command and status processing), plus the adapter driver path length.
- iSCSI Service in host (over an Ethernet Link Service in NIC) - The Linux host path length for an iSCSI device driver transaction, plus the Linux host overhead for performing the associated TCP/IP processing.
- iSCSI Service offloaded to iONIC - The Linux host path length for an iSCSI device driver transaction, plus the Linux host path length performing the associated RDMA Service Queue Pair processing.

All else being equal (e.g. link speeds), the “iSCSI Service in host” case is much less efficient than a parallel SCSI link. This may help explain why volume of shipments for the first generation of iSCSI subsystems was relatively small. The “iSCSI Service in iONIC” case is shown as a band, because its efficiency depends to a great extent on the implementation’s function split. A good implementation is one where the code path associated with setting up the iSCSI data transfer matches the parallel SCSI adapter’s code path length. A good implementation requires the RDMA Service to support: submission of multiple work requests at one time, fast registration, and send with invalidate. The resulting path length is still higher than the path length associated with a parallel SCSI driver, but not by an order of magnitude.

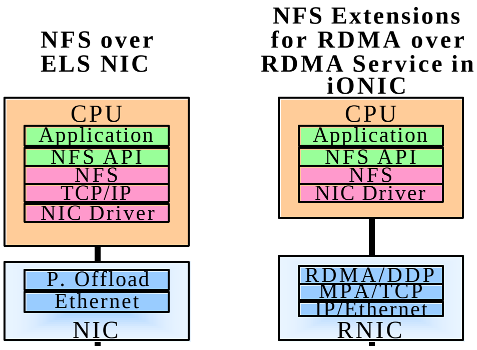
The second inhibitor for IP/Ethernet deployment as a storage network is switch latencies. Currently Ethernet switches have horrendously long latencies compared to FC switches. The 1 Gigabit/s Ethernet generation had switch latencies from 10 to 50 microseconds, compared to under .5 to 2 microsecond latencies for Fibre Channel switches. For a multi-hop switch network, the difference in switch latencies becomes a major issue for performing storage over FC.

Several Ethernet switch vendors are focusing on closing this gap. For example, at Hot Chips 13 Nishan [27] described their Tyrant IP switch as having under 2 us switch latencies. From a pure technical standpoint there really is no reason for the switch latency delta that has existed in the past between Ethernet and FC. As a result, in the near future, as Ethernet switch vendors pursue the lucrative market opportunity associated with iSCSI storage, the gap between Ethernet and FC switch latencies will likely close.

7.1.2File Mode access over Ethernet

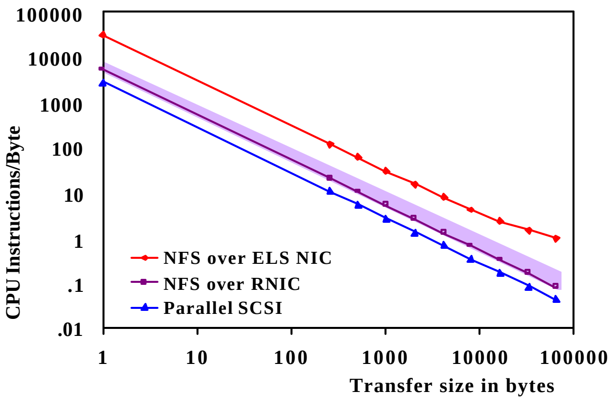
Figure19 compares the functions performed by a traditional networked file system to the functions performed by a networked file system that uses an RDMA Service and offloads the network stack to the adapter. Examples of such an NFS RDMA Service are Direct Access File System (DAFS) and the three internet-drafts [34] [35] [36] being pursued in the IETF to provide NFS extensions for RDMA.

Figure 19. File mode storage access options



Again, I am not at liberty to provide measurements for the three cases in Figure20, so the three cases are based on code path estimates.

Figure 20. File mode I/O vs SCSI overheads



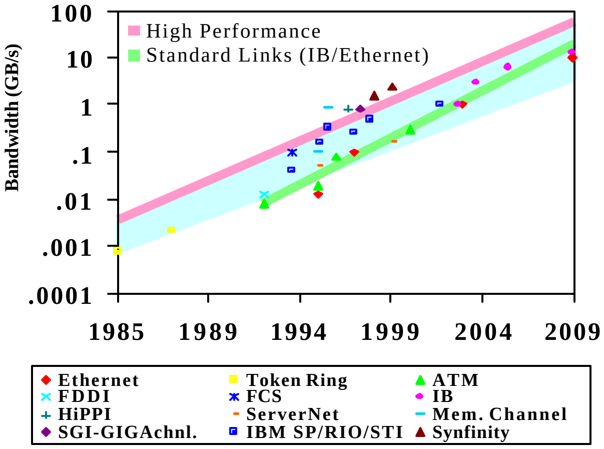
- Parallel SCSI - Same as the block mode case.
- NFS over ELS (Ethernet Link Service) NIC - The NFS over an Ethernet Link Service NIC on Linux.
- NFS over RNIC - The Linux host path length for NFS over an RDMA Service that has been offloaded to an RNIC or multiple-service iONIC.

As internet Offload NICs become available with an RDMA Service, the gap between the block mode I/O (parallel SCSI) and file mode I/O (e.g. NFS) will become much smaller. Once again, mileage will vary by implementation, but the significant reduction in the performance gap will represent a major discontinuity for customers of networked storage.

8. CLUSTER NETWORKS

Cluster networks include the adapters, links, and switches that support message passing between servers in a cluster. Cluster network span a wide range of application environments. The most demanding of these application environments require low-latency, high-bandwidth data transfers with low processor overheads and require clusters that can scale to thousands of servers. Figure21 illustrates the growth in cluster interconnect bandwidth over the past 12 years. As can be seen in Figure21, clusters use a mix of industry-standard and proprietary cluster network technologies.

Figure 21. Cluster interconnect performance growth



Industry standard cluster interconnects are available from several vendors, and include Ethernet and, more recently, InfiniBand. Ethernet is commonly used for clusters emphasizing low cost and simplified management. InfiniBand is emerging for systems that require higher performance.

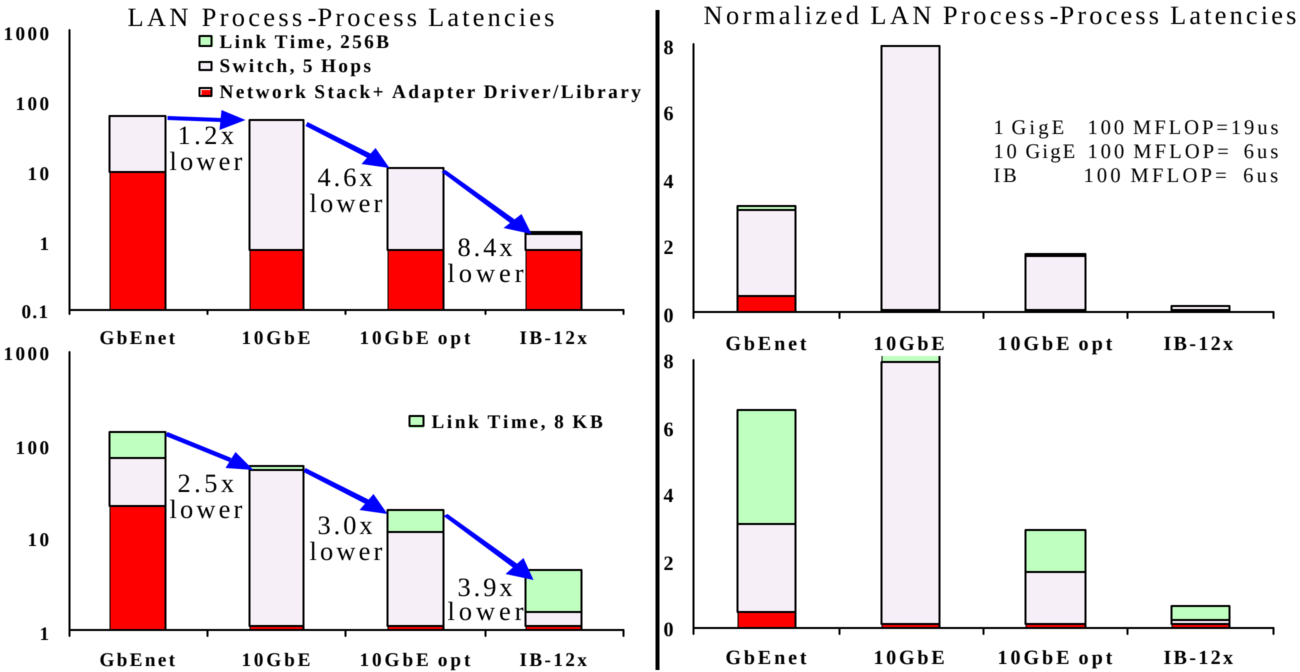
Proprietary cluster interconnects either target server platforms from multiple vendors or from a single vendor. Proprietary cluster interconnects that target multiple platforms attach through PCI, and include: Myrinet, at 250 MByte/sec, and Quadrics, at 340 MByte/sec. Proprietary cluster interconnects that attach to a single server typically attach to a higher bandwidth, lower latency interface, and include: IBM (SP Switch), HP (Memory Channel), SGI/Cray (CrayLink), and NEC (Earth Simulator network). The proprietary market has been lucrative enough to allow some of the proprietary interconnects to be enhanced over several generations.

In the past, standard cluster interconnects used sockets over an Ethernet Link Service NIC, which has comparably low-bandwidth and high-overhead. Whereas proprietary cluster interconnects have used low-overhead, proprietary communication interfaces. Over the past several years, some of the functions available on proprietary cluster networks have been standardized by IB, and more recently RNICs.

In the past, standard cluster interconnects attached through adapters that were themselves connected to standard busses (e.g. PCI). As covered earlier, I/O attachment through PCI has several issues related to MMIO operations. In the future, to remove these issues, some servers will likely attach standard cluster interconnects much closer to the CPU.

As IB and RNIC infrastructure gets deployed and matures, proprietary interconnects will be pushed to the extreme high end of the market. Though both IB and RNICs provide the necessary CPU overhead reductions (see earlier sections), bandwidth and switch latencies will, for the near future, be a key distinguishing characteristic for IB when compared to Ethernet. Figure22 is derived from an analytical model that compares the process-process latencies of Ethernet and IB links.

Figure 22. Reduction in cluster process-process latencies



Late 2003 to 2004 interconnects	**1 Gig Ethernet**	**10 GigE (Today)**	**10 GigE Optimized**	**IB 12x**	**Myrinet**	**Quadrics**	**IB 4x**
Switch Latency (ns)	**10000 to 20000**	**10000**	**2000**	**100 to 200**	**150**	**35**	**160+**
Link Bandwidth (MB/s)	**100**	**1024**	**1024**	**3072**	**250**	**400**	**1024**

The move from TCP/IP to more efficient mechanisms (e.g., RDMA) imported from proprietary networks will dramatically improve the performance of 10 GigE in both latency and host CPU overhead. The main difference between IB and Ethernet are the higher link bandwidths possible with IB (3 GB/s vs 1 GB/s) and the lower switch latencies (100 ns or lower vs several microseconds).

The model used in Figure22 depicts the following cases (all assume no link congestion):

- 1 GigE - The network stack and driver latency is based on the Linux measurements described earlier in this document. The switch latency is based on a 5 hop, cluster fabric built using generally available Gigabit Ethernet switches with 10 us latencies.
- 10 GigE Now - The network stack and driver latency is based on an RNIC approach that offloads the network stack and removes receive side copy overheads. The switch latency is based on a 5 hop, cluster fabric built using generally available 10 Gigabit Ethernet switches, each having 10 us latency.
- 10 GigE Opt(imized) - The network stack and driver latency is based on an RNIC approach that offloads the network stack and removes receive side copy overheads. The switch latency is based on a 5 hop, cluster fabric built using possible next generation 10 Gigabit Ethernet switches, each having 2 us latency [27].
- 12x IB case - The network stack and driver latency is based on an IB HCA approach that offloads the network stack and removes receive side copy overheads. The switch latency is based on a 5 hop, cluster fabric built using IB switches, each having 100 ns latency, which is reasonable considering the 4x switch generation now available achieves 160 ns switch latencies [18].
- For comparison purposes, the table includes Myrinet [8], Quadrics[30], and IB 4x fabrics.

The two bar charts on the left in Figure22 represent the total process-process communication latency in microseconds for a 256 byte transfer (top) and an 8 KB transfer (bottom). The two bar charts on the right in Figure22 represent the contribution of each major element in the cluster (link, switch, and host communication software stack). The normalization is based on the time it takes to execute 100 MFLOPs on a processor available at the time the link technology was introduced. That is, at the time Gigabit Ethernet was introduced a processor was able to perform 100 MFLOPs in 19 microseconds. For IB and 10 GigE, 6us was used.

As can be seen in Figure22, an IB based cluster network provides 29x to 46x lower latency than a Gigabit Ethernet cluster. Compared to today's 10 Gigabit Ethernet, IB provides 12x to 39x lower latency. In the future, 10 GigE will likely close that gap to between 3.9x and 8.4x. However, IB is not standing still and will likely retain the gap through the deployment of higher bandwidth links (e.g. 12x 500 MB/s or 12x 1 GB/s) and even lower latency switches.

9. SUMMARY (OF AUTHOR'S VIEWS)

I/O adapters for servers of all types will likely attach through the high-volume PCI family, including PCI Express, because of PCI's low cost and simplicity of implementation. For the next three years, most vendors will continue using the parallel PCI busses, such as PCI-X, PCI-X2.0 DDR, and PCI-X2.0 QDR. PCI Express will first appear in desktop systems as a replacement for the Advanced Graphics Port and as an ASIC interconnect. By 2006 or 2007, servers will likely use PCI Express adapters to attach high-bandwidth I/O. Servers may also use InfiniBand to attach high-end I/O subsystems, such as a high-end storage server.

I/O expansion networks will likely use proprietary link architectures or proprietary implementations of standards, such as InfiniBand and PCI Express. Although vendors of high-end servers will simply extend their proprietary networks in the near term, some vendors will migrate to InfiniBand by 2005 or 2006. These vendors will provide bridges to standard I/O adapters using PCI-X or PCI Express. Lower-end vendors will likely use PCI Express or PCI Express with extensions as an I/O expansion network.

Clusters for commercial applications and most high-performance computing applications will continue using Ethernet networks because they will satisfy most requirements for cost, latency, scalability, processor and memory overheads, and virtualization. High-end clusters will likely use Ethernet or InfiniBand physical links with custom adapters and high-performance switches. Ethernet RNICs will be used in the price/performance end of the market, while InfiniBand will be used when high bandwidth is an absolute requirement. Clusters with the most demanding performance requirements for high-performance computing will likely use standard InfiniBand networks or InfiniBand networks with proprietary extensions.

For Local Area Networks, adapter vendors will apply InfiniBand techniques (i.e. RDMA Service) to offload IP processing onto I/O adapters using TCP/IP offload engines and RNICs. The first generation of iONICs that offer a TOE Service is available now. Although the TOE Service will not completely eliminate receive-side copies, they support communication with clients that do not provide an RDMA Service. iONICs with an RDMA Service will eliminate receive-side copies through the use of RDMA Read and RDMA Writes. The chip manufacturing cost for a TOE or RDMA Service will be similar to that of an InfiniBand host channel adapter because they support a comparable level of function.

Storage area networks will increasingly use Ethernet with the iSCSI and iSER protocols. Although some vendors will initially design adapters with a specialized iSCSI mechanism for data placement, eventually, all vendors will likely use standard RDMA operations as a general-purpose mechanism for placing data. Fibre Channel use will continue, even though the 10Gbit/s generation of iSCSI will offer comparable performance.

The spectrum of offload adapter designs (IB HCAs and iONICs) will range from approaches that implement critical data path functions in state machine to approaches that implement all functions in code. As a result, offload adapter performance will vary by vendor.

10. ACKNOWLEDGEMENTS

I'd like to thank Todd Boyd, Herman Dierks, Janice Girouard, and Erich Nahum for measurements taken on IBM servers and subsequent instruction trace analysis. Also to be thanked is Mark Johnson, thanks for grammatical comments. And of course all the folks I have worked with in the InfiniBand Trade Association, RDMA Consortium, and, more recently, IETF.

Finally, the trends and directions in this paper represent my views and do not imply the view or directions of: IBM or any of the standards groups I am directly or indirectly associated with (IBTA, RDMAC, IETF, and PCI-SIG.

11. REFERENCES

[1] ASCI Purple Statement of Work, Lawrence Livermore National Laboratory, http://www.llnl.gov/asci/purple/Attachment_02_PurpleSOWV09.pdf

[2] Alaska-X, 10 GBase-CX4 Transceiver 88x2088, from Marvell. https://www.marvell.com/products/transceivers/alaskax/PRODUCT%20BRIEF--Alaska%20X--88X2088%20%20MV-S100529-00%20Rev.%20B.pdf.

[3] Balaji, P.; Shivam, P.; Wyckoff, P.; Panda, P. "High Performance User Level Sockets over Gigabit Ethernet" in Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER 2002).

[4] Boyd, W.; Recio, Renato. "I/O Workload Characteristics of Modern Servers" in IEEE Workshop on Workload Characterization: Methodology and Case Studies Nov. 1988.

[5] Buonadonna, P.; Culler, D.; "Queue pair IP: a hybrid architecture for system area networks" in Proceedings. 29th Annual International Symposium on Computer Architecture, 2002.

[6] Camarda, P.; Pipio, F.; Piscitelli, G. "Performance evaluation of TCP/IP protocol implementations in end systems in Computers and Digital Techniques, IEEE Proceedings - Volume: 146 Issue: 1, Jan. 1999.

[7] Chase, J.; Gallatin, A.; Yocum, K. "End System Optimizations for High-Speed TCP" in IEEE Communications Magazine - April 2001.

[8] Chen, H.; Wyckoff, P. "Performance evaluation of a Gigabit Ethernet switch and Myrinet using real application cores" in Hot Interconnects 8 August 2000.

[9] Clark, D.D.; Romkey, J.; Salwen, H. “An analysis of TCP processing overhead” in Proceedings of the 13th Conference on Local Computer Networks, 10-12 Oct. 1988.

[10] Common Electrical I/O an Optical Internetworking Forum white paper (on a new project to define new electrical specifications for 4.976 to 6+ Gigabit and 9.95 to 11+ Gigabit signaling) at http://www.oiforum.com/public/documents/CEIWP.pdf).

[11] Hilland, J.; Culley, P.; Pinkerton, J.; Recio, R. RDMA Protocol Verbs Specification (Version 1.0). http://www.rdmaconsortium.org/home/draft-hilland-iwarp-verbs-v1.0-RDMAC.pdf.

[12] Hoke, J.; Bond, P.; Livolsi, R.; Lo, T.; Pidala, F.; and Steinbrueck. G. IBM eServer z900 in IBM Journal of Research and Development, Volume 46, Numbers 4/5, 2002. http://researchweb.watson.ibm.com/journal/rd/464/hoke.html

[13] Horst R. and Garcia, D; ServerNet SAN I/O Architecture in Hot Interconnects V, 1997. http://www.cs.berkeley.edu/~culler/hoti97/horst.ps.

[14] HP Hyperfabric and Hyperfabric 2. http://www.hp.com/products1/unixserverconnectivity/adapters/adapter06/infolibrary/prod_brief_final_may2002.pdf.

[15] IBM eServer pSeries SP Switch and SP Switch2 Performance, Version 8. February, 2003. http://www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/sp_switch_perf.pdf.

[16] IEEE Standard 802.3ae ™ - 2002; Amendment: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation http://standards.ieee.org/getieee802/download/802.3-2002.pdf.

[17] InfiniBand™ Architecture Specification Volume 1, Release 1.1. November 6, 2002. http://www.infinibandta.org/specs.

[18] InfiniSwitch “Fabric Networks 1200 and 2400 Switches”, http://www.fabricnetworks.com/dl.phtml?id=309.

[19] Interconnect Software Consortium. http://www.opengroup.org/icsc/.

[20] InterNational Committee for Information Technology Standards’s T11 Workgroup. http://www.t11.org/index.htm

[21] Kay, J.; Pasquale, J. “Profiling and Reducing Processing Overheads in TCP/IP” in IEEE/ACM Transactions on Networking, Vol. 4, No. 6, December 1996.

[22] Kim, K.; Sung,H.; Lee, H. “Performance analysis of the TCP/IP protocol under UNIX operating systems for high performance computing and communications” in High Performance Computing on the Information Superhighway, 1997.

[23] Laudon, J.; Lenoski, D; System overview of the SGI Origin 200/2000 product line in Compcon '97. Proceedings, IEEE, 23-26 Feb. 1997.

[24] Lee, J.; Razavi, B. “A 40 Gb/s clock and data circuit in .18um CMOS Technology” in 2003 IEEE International Solid-State Circuits Conference.

[25] Mathis, H.; McCalpin, J.; Thomas, J. IBM Systems Group white paper - IBM eServer pSeries 690. http://www-1.ibm.com/servers/eserver/pseries/hardware/whitepapers/p690_config.pdf

[26] Meet the HP Superdome Servers. May 2002, a white paper from Hewlett-Packard. http://www.hp.com/products1/servers/scalableservers/superdome/infolibrary/technical_wp.pdf

[27] Oberman,S.; Mullendore, R.; Malik, K.; Mehta, A.; Schakel, K.; Ogrinc, M.; Mrazek, D. Tyrant: A High Performance Storage over IP Switch Engine in Hot Chips 13, August 2001. http://www.hotchips.org/archive/hc13/hc13pres_pdf/16nishan.pdf.

[28] PCI-X Addendum to the PCI Local Bus Specification Revision 1.0. September 22, 1999. http://www.pcisig.com/specifications/pcix_20/pci_x.

[29] PCI Express Base Specification Revision 1.0. April 29, 2002. http://www.pcisig.com/specifications/pciexpress.

[30] Petrini, F.; Feng, W.; Hoisie, A.; Coll, S.; Frachtenberg, E. The Quadrics network (QsNet): high-performance clustering technology in IEEE Micro, 2/2002. http://www.quadrics.com/website/pdf/ieeemicro_feb_2002.pdf.

[31] Recio, R. RDMA enabled NIC (RNIC) Verbs Overview. http://www.rdmaconsortium.org/home/RNIC_Verbs_Overview.pdf.

[32] Recio, R.; Culley, P.; Garcia, D.; Hilland, J. An RDMA Protocol Specification (Version 1.0). http://www.rdmaconsortium.org/home/draft-recio-iwarp-rdmap-v1.0.pdf.

[33] Shaeffer, D.; Tao, H.; Lee, Q.; Ong, A.; Condito,V.;Benyamin, S.; Wong, W.; Si, X.; Kudszus, S.; Tarsia, M. “A 40/43 Gb/s Sonet OC-768 SiGE 4:1 MUX/CMU” in 2003 IEEE International Solid-State Circuits Conference.

[34] Talpey, T.; Callaghan, B. “NFS Direct Data Placement” at http://www.ietf.org/internet-drafts/draft-callaghan-rpc-rdma-00.txt.

[35] Talpey, T.; Callaghan,B. “RDMA Transport for ONC RPC” at http://www.ietf.org/internet-drafts/draft-callaghan-rpcrdma-00.txt.

[36] Talpey, T.; Shepler, S. “NFSv4 and Sessions Extensions” at http://www.ietf.org/internet-drafts/draft-talpey-nfsv4-rdma-sess-00.txt.

[37] VC 1032, Quad 3.2 Gb/s XAUI SerDes (with Redundancy) for 10Gigabit Ethernet Networks. http://www.velio.com/downloads/VC1062_VC1061_Product_Summary.pdf

[38] Yen, J.; Case, M.; Nielsen, S.; Rogers, J.; Srivastava, N.; Thiagarajah, R. “A fully integrated 42.3 Gb/s clock and data recovery and 1:4 DEMUC IC in InP HBT Technology” in 2003 IEEE International Solid-State Circuits Conference.