Using TCP/IP Offload Engine and Receive-Side Scaling to Improve Performance and Efficiency on HP ProLiant Servers in Microsoft® Windows® Environments

Technology Brief

Abstract

This paper describes how TCP/IP Offload Engine (TOE) reduces server CPU cycle demands and increases application scalability by shifting the communications protocol stack (TCP/IP) from the host CPU to the network adapter. This paper also discusses Receive-side Scaling (RSS), which balances the incoming traffic across all the processors in a multi-processor server. Microsoft's Scalable Networking Pack is required for both TOE and RSS. These technologies are supported on recently-announced HP ProLiant servers, on HP BladeSystem servers, as well as on HP networking server adapters and blade mezzanine cards.

Introduction

Increasing demands for improved networking performance have been satisfied by increased networking speeds, from 10Mbps Ethernet in the mid-1970s to 10 Gigabit Ethernet today. But faster protocol speeds have not solved all the problems for servers. For example, protocol processing can sometimes become a bottleneck because the overhead involved can put undue burden on a server's CPU.

The release of Microsoft's Scalable Networking Pack now makes TCP/IP Offload Engine (TOE) a readily-available technology for relieving the server CPU from processing the protocol stack. TOE offloads the protocol processing from the host to the server adapter (also known as a NIC), thereby freeing CPU cycles for other duties. With TOE, network communications are improved and server efficiency is increased.

Scalable Networking Pack also enables Receive-side Scaling (RSS). Currently, the TCP/IP stack allows only one CPU to handle incoming traffic, a condition that doesn't lend itself to scaling well on multi-processor servers. A single CPU can be completely taxed with processing the incoming traffic while the remaining CPUs are virtually idle. RSS works to solve this situation by dynamically load balancing the incoming traffic across all the processors in a server.

TCP/IP Offload Engine (TOE)

Transmission Control Protocol/Internet Protocol (TCP/IP) is the suite of protocols that drive the Internet. Every computer connected to the Internet must use the protocol to send and receive information, which is transmitted in fixed data block (packet) sizes so that heterogeneous systems can communicate in a standardized format. Computers implement the TCP/IP protocol stack to process outgoing and incoming packets.

The TCP/IP Problem

Today, TCP/IP stack implementations are usually found in operating system software and, therefore, must be processed by the CPU. This is not a problem so long as CPU cycles are sufficient to handle both the protocol work and the applications. However, as network speeds increase, servers are not able to keep up, sometimes requiring the server CPU to devote more processing power to the TCP/IP stack than to the applications on the server.

Why does TCP/IP protocol processing require so much CPU power? The TCP/IP stack of protocols was developed to be a language for all types of computers to transfer data across different physical media. TCP/IP protocols involve over 70,000 software instructions that provide all the necessary reliability mechanisms, error detection/correction, sequencing, recovery, and other communications features. As a result, protocol processing of incoming and outgoing network traffic consumes CPU cycles that could be used for software applications. This also negatively impacts an application's ability to scale across a large number of servers. This situation has led to the widely-accepted estimate that it takes a Gigahertz of CPU clock speed to drive a Gigabit of Ethernet networking at wire speed.

Checksum Offload and Large Send Offload (LSO)

The embedded networking adapters on ProLiant servers and stand-alone HP networking adapters have traditionally supported two simple offload systems, TCP/IP checksum offload, and Large Send Offload (LSO). Both of these offload technologies help reduce CPU utilization by moving CPU-intensive calculations to the network adapter.

TCP checksum offload moves the process of calculating the TCP and IP checksum packets from the server CPU to the server adapter. While TCP checksum offload reduces the load on the server processor, it results in only modest overall improved system response.

LSO, also known at TCP/IP segmentation offload, takes the job of segmenting data into network frames and moves it from the TCP/IP stack on the host to the network adapter. When large packets are offloaded to the network adapter, it is the adapter that segments the data into smaller pieces to be sent over the network. This not only decreases CPU utilization, but it also improves network performance. LSO is most efficient with large packets, and, as its name indicates, is useful only for network traffic that is being sent; in order to derive the benefits of LSO, the operating system, the network adapters, and the network driver must all support this feature.

What's needed to run TOE

To implement TOE (and Receive-side Scaling, as well), four separate elements are required: two from Microsoft and two from HP:

- Microsoft® Windows Server™ 2003, Service Pack 1, or later
- Microsoft Windows Server 2003 Scalable Networking Pack, available directly from Microsoft through www.microsoft.com/snp
- HP ProLiant server with embedded multifunction network adapter or stand-alone NC-series multifunction server adapter
- HP networking driver The HP networking driver that supports TOE Chimney. This driver is currently identified as "HP NC370x/371x/373x/374x/380x Multifunction Gigabit Server Adapter Driver for Windows Server 2003 version 2.6.2.0."

  The driver is available as release version NCDE 8.37 of the "HP Network Server Adapters and Upgrade Modules Software and Documentation" CD, in SmartStart version 7.51 (or later), and also from the Software and Driver Download page at www.hp.com.

How TOE Works

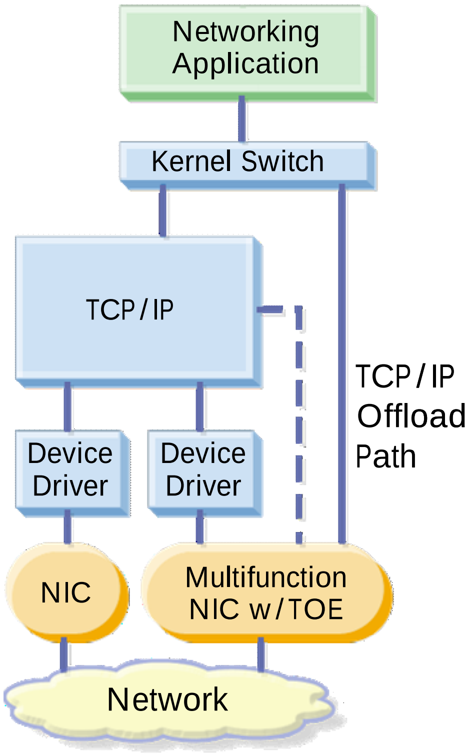

Figure 1: TCP/IP Offload Path

Figure 1 is a graphical representation contrasting the function of a traditional adapter with that of an adapter supporting TOE. In the path for standard network traffic (shown as passing from the network through the NIC and device driver), data moves from the hardware through the TCP/IP protocol stack to the kernel switch and then on to the application. In this scheme, the host CPU is responsible for processing all the network transactions, including numerous interrupts, event notifications, multiple memory copies, packet segmentation and reassembly, and acknowledgements. CPU cycles must also be dedicated to buffering packets so that they can be processed with a TCP connection. All of this puts a heavy load on the host CPU.

By contrast, the data path offloaded traffic (shown passing from the network through the multifunction NIC with TOE up to the kernel switch) indicates that the protocol stack is offloaded to a TOE-enabled hardware adapter supporting Microsoft Windows Server 2003 Scalable Networking Pack. The protocol stack processing is done by a performance-optimized ASIC in combination with a networking driver that supports TOE. The benefits of offloading the TCP/IP protocol stack include lowering CPU utilization by reducing interrupts and context switches, as well as reducing the number of memory copies required for each transaction.

Performance Improvements with TOE

The following graphs illustrate the efficiencies gained by using an HP multifunction network adapter. In the first, throughput achieves near wire speed when packets are above 4KB, and CPU utilization, which is about 35% with the smallest packet size, shows only a modest improvement with the largest packets. In this scenario, there is no offload of any kind.

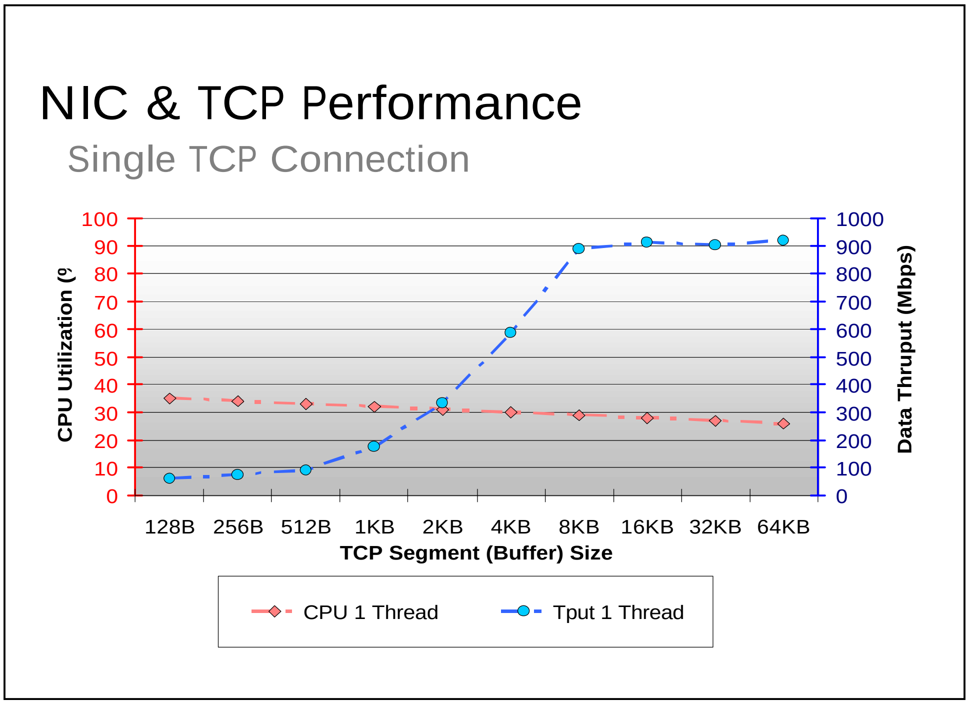

Figure 2: Single TCP Connection, non-TOE CPU Utilization and Throughput

In Figure 3 below, lines showing multiple threads have been included with the lines from Figure 2, which show a single thread. In the most dramatic difference, multiple threads completely tap out the CPU utilization when small packets are transmitted; furthermore, near wire speed is achieved with 2KB packets. LSO reduces CPU utilization as larger payloads are transmitted. Note that with the largest packets, CPU utilization and throughput with multiple threads are about the same as those with a single thread.

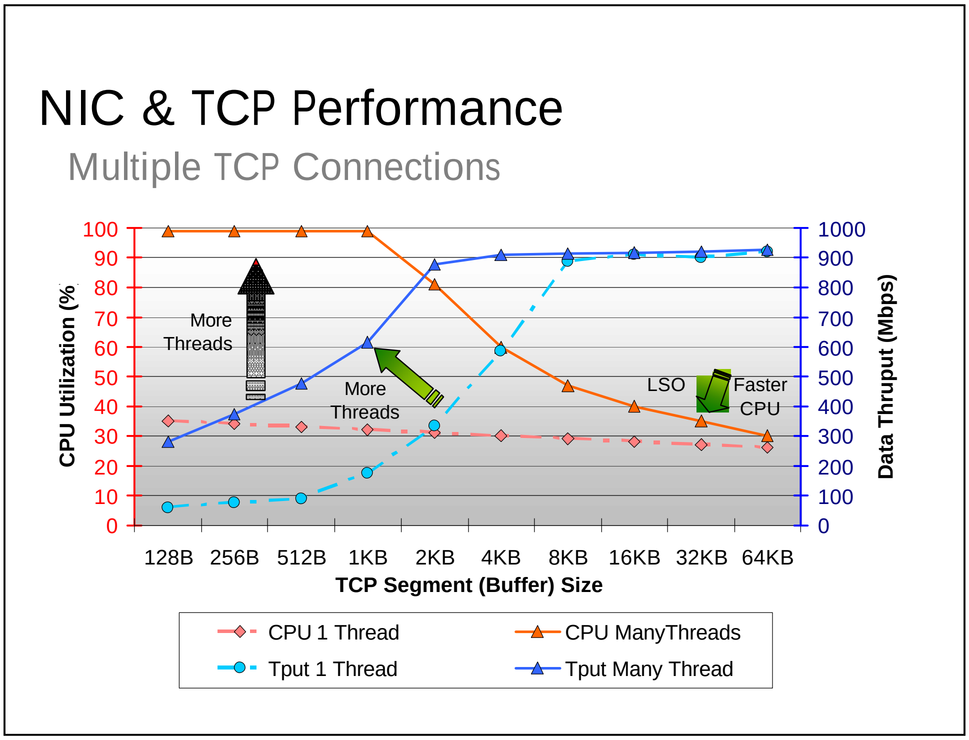

Figure 3: Multiple TCP Connections, non-TOE CPU Utilization and Throughput

Finally, in Figure 4, the lines representing multiple TCP/IP Offload connections have been compared to those from the non-TOE multiple connections represented above in Figure 3.

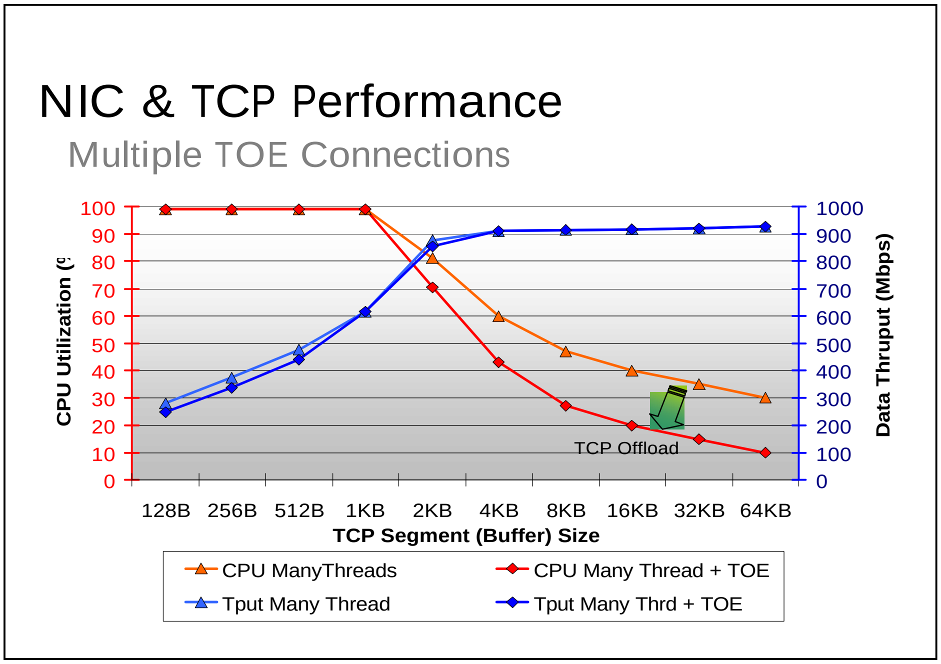

Figure 4: Multiple TCP Offload Connections – CPU Utilization and Throughput

The single biggest difference relates to the CPU utilization with the largest packet sizes. By using Microsoft Windows Server 2003 Scalable Networking Pack with an HP multifunction server adapter, the protocol stack is offloaded to the adapter, thereby reducing CPU utilization from 30% to10%.

This is a significant point: TCP/IP Offload Engine is primarily about improving server efficiencies, not about increasing wire speeds. The throughputs indicated in these graphs are remarkably similar; it is the lower CPU utilization that is the key advantage.

There is a common misunderstanding that when a server is configured with TCP/IP Offload Engine, it offloads every connection. This is not the case. Only long-lived connections are loaded to the network adapter, and the Windows operating system decides. Additionally, TOE applies only to TCP connections; therefore, any other protocol (such as UDP) is not offloaded and must be processed in the normal OS stack.

For protocol offload to be advantageous, the Windows operating system must decide that the connection will gain enough efficiency so as to make the offload connection worth doing. The life of the TCP connection depends solely on the conversation taking place at the TCP level. Once an offload decision has been made, the host identifies TCP and IP states, which are then passed on to the adapter. While this process takes place data is being buffered, so if the packets are small and the connections are brief, it does not pay to offload the protocol.

How TOE helps

High-speed networks benefit most when the TCP/IP stack is offloaded to a dedicated network adapter supporting TOE, where the TOE logic is embedded in a chip or firmware on the network adapter that uses a customized ASIC.

With Scalable Networking Pack and HP Multifunction Server Adapters running TOE, ProLiant server efficiencies increase. Additionally, this standards-based technology does not require any change to existing applications, thereby protecting server investments while also putting into place networking capability that scales from multiple Gigabit connections to the emerging 10Gigabit Ethernet products that are currently being deployed.

Receive-side Scaling (RSS)

While TCP/IP Offload Engine is best used for long-lived connections, Receive-side Scaling is designed for short-live connections. RSS is best thought of as load-balancing for incoming traffic.

What's needed to run RSS

To implement RSS, four separate elements are required: two from Microsoft and two from HP:

- Microsoft® Windows Server™ 2003, Service Pack 1, or later
- Microsoft Windows Server 2003 Scalable Networking Pack, available directly from Microsoft
- HP ProLiant server with embedded multifunction network adapter or stand-alone NC-series multifunction server adapter
- HP networking driver The HP networking driver that supports TOE Chimney. This driver is currently identified as "HP NC370x/371x/373x/374x/380x Multifunction Gigabit Server Adapter Driver for Windows Server 2003 version 2.6.2.0."

  The driver is available as release version NCDE 8.37 of the "HP Network Server Adapters and Upgrade Modules Software and Documentation" CD, in SmartStart version 7.51 (or later), and also from the Software and Driver Download page at www.hp.com.

How Receive-side Scaling helps

With Receive-side Scaling, incoming short-lived traffic will be balanced across multiple processors while preserving the ordered delivery of packets. Additionally, Receive-side Scaling allows the incoming traffic to be dynamically adjusted as the system load varies. As a result, any application with heavy networking traffic running on a multi-processor server will benefit. RSS is independent of the number of connections, allowing it to scale well. This will make RSS particularly valuable to web servers and file servers handling heavy loads of short-lived traffic.

For Further Information

For additional details about HP products mentioned in this technology brief, refer to:

Resource Description	Web address
HP ProLiant and BladeSystem servers	www.hp.com/go/proliant
HP Networking Products	www.hp.com/servers/networking