



OS Bypass:

*Key to Eliminating Overhead
in Ethernet Networks*

by Kris Meier

Introduction

The demand for higher speed server interconnects in support of clustering, storage networks, and sheer bulk data movement continues to drive Ethernet's evolution. In transitioning from 1 gigabit to 10 gigabit per second data rates, Ethernet is poised to handle the most demanding data center applications in all three of these areas. However, taking full advantage of this 10x increase in performance requires the elimination of all three elements of Host CPU overhead related to networking: buffer copies, transport processing, and application context switches.

Remote Direct Memory Access (RDMA) allows a network adapter to place data directly into the memory of a remote server without the typical buffer copies. This saves precious memory bandwidth and CPU cycles and eliminates approximately 20% of the Host CPU overhead¹.

Transport Offload Engines (TOEs) perform TCP stack processing on the network adapter. This removes the transport processing burden from the CPU and eliminates approximately 40% of additional Host CPU overhead.

The remaining 40% of Host CPU overhead is attributed to application context switches. Context switches occur when process execution moves from user space to kernel space. Of the three sources of networking overhead, context switches have been discussed the least and warrant further consideration.

Context Switch

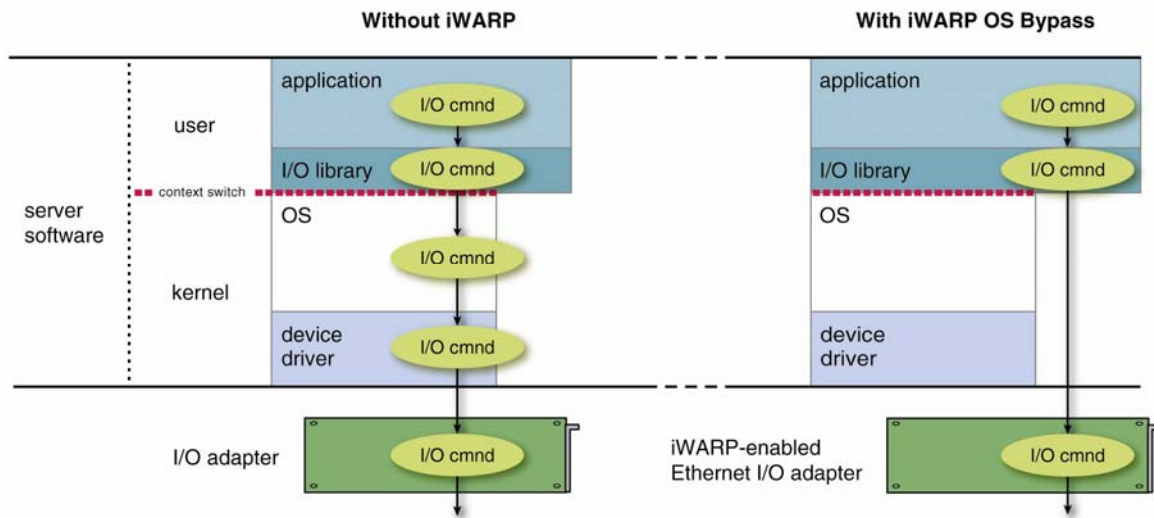
Simply put, user space is where all user programs execute. Historically, applications operating in user space make system calls into the kernel for privileged operations such as input/output commands to networking or storage devices.

Kernel space is where the OS, device drivers and hardware interrupt handlers run. The kernel provides a safe interface to hardware, provides inter-process security, gives different processes a fair share of the resources, and arbitrates access to resources/hardware.

Transitions from user to kernel space (and the reverse) have historically been required to pass data between user programs and their clustering, storage, and networking hardware resource. Each transition requires saving the user process context data and loading the kernel context data. The process of saving the user process information and loading the kernel process information is known as the context switch. Typically a context switch involves saving the address space and software stack information, as well as the register set (program counter, stack pointer, instruction register and other general processor registers) from the current process and loading the corresponding information for the new

process. With this information the CPU begins execution of the kernel process utilizing the restored registers and address space.

The overhead of saving and restoring context information fundamentally limits application I/O performance. As mentioned above, in the case of TCP/IP user to kernel transition, it can account for approximately 40% of the Host CPU networking overhead. The technique for eliminating the user to kernel transition and its associated context switch is known as *OS bypass*. As shown in the diagram, operating system calls are avoided by updating the I/O library to take advantage of OS bypass capabilities. This modification is transparent to applications and enables direct communication of all commands to the I/O adapter, eliminating the user to kernel transition. OS bypass is a well-proven technique that has been used for years in the highest performance cluster interconnects.



iWARP's OS bypass eliminates expensive calls to the OS kernel typically required for context switching.

Elimination of Overhead

Recognizing the Host CPU overhead problem, the RDMA consortium (www.rdmaconsortium.org) and IETF (www.ietf.org) have developed a set of standard extensions to Ethernet and TCP/IP that eliminate all three sources of overhead. These specifications are collectively called iWARP. The iWARP specification enables an iWARP-compliant Ethernet Channel Adapter (ECA) to transparently replace Ethernet NICs i.e. an ECA is completely compatible with today's Ethernet infrastructure – cables, switches/routers, and applications. Additionally, iWARP defines a new interface that enables the application software to interface directly with the ECA. This provides additional benefits to those applications demanding the highest levels of performance. To fully eliminate Host CPU overhead an iWARP ECA must support TCP offload, RDMA, and OS bypass – anything less will consume CPU resources as network load increases.

If Ethernet can, Ethernet will

iWARP is the next step in the evolution of Ethernet. With the increase in wire speed, elimination of network overhead and compatibility with today's Ethernet infrastructure, iWARP Ethernet has broad market appeal in all three data center connectivity applications.

¹ Based on laboratory tests run at NetEffect Inc.

Kris Meier is the Product Manager for NetEffect's line of iWARP Ethernet Channel Adapters. He can be reached at KMeier@NetEffect.com