Technology Brief

Intel® I/O Acceleration Technology

Intel® I/O Acceleration Technology

Move network data more efficiently through the new Dual-Core Intel® Xeon® processor-based servers for fast, scalable, and reliable networking.

Today, more than ever, business success relies on rapid transfer and processing of data. To improve data transfer to and from applications, IT managers continue investing in new networking and storage infrastructure to achieve higher performance. However, network I/O bottlenecks have emerged as the key IT challenge in realizing full value from server investments. Intel® I/O Acceleration Technology (Intel® I/OAT), available in the new Dual-Core Intel® Xeon® processor-based servers, addresses the primary causes of network I/O bottleneck issues. Intel I/OAT moves network data more efficiently through new Dual-Dore Intel Xeon processor-based servers for fast, scalable, and reliable networking. It provides network acceleration that scales seamlessly across multiple Ethernet ports and is a safe and flexible choice for IT managers due to its tight integration into popular operating systems.

Historical Perspective

Historically, network infrastructure investments included migration from Fast Ethernet to Gigabit Ethernet (GbE). In addition to this investment in greater bandwidth, IT has invested in a variety of system and architectural enhancements such as redundant arrays of independent disks (RAID) and multi-tiered data centers. Throughout these evolutionary stages, network server performance kept pace with network traffic increases. However, with the increasing number of network data appliances combined with the increase in network applications, servers must manage an unprecedented data load, which threatens to outpace server processing capabilities. As a result, IT managers are now asking, "After investing in a 10X improvement in network bandwidth, why aren't we seeing comparable improvements in application response time and reliability?"

The answer is surprisingly simple. Since the 1980s, despite significant innovative technology leaps, the method for processing TCP/IP (the network stack) has remained largely unchanged. This results in significant system overhead as the CPU must process the non-optimized TCP protocol.

Add to the system overhead challenge the issue of inefficient memory access related to network data processing, and significant network I/O bottlenecks emerge throughout the system. This results in inefficient CPU utilization and limited network throughput.

Solving these challenges requires more than just point solutions that address only incremental deficiencies. Instead, a platform-wide approach is necessary to address the bottlenecks evident throughout a server. To provide such a solution, Intel research and development teams evaluated the entire server architecture to determine platform-level advances that would facilitate higher GbE and 10 Gigabit Ethernet (10GbE) rates while addressing the growing trends seen in network environments (network storage, network applications, and so on). Their response: Intel I/OAT—a server platform technology that moves network data more efficiently through new Dual-Core Intel Xeon processor-based servers for fast, scalable, and reliable networking.

The Growing Server I/O Traffic Jam

Understanding the importance of Intel I/OAT requires an understanding of the real nature of the problem being addressed. This is best done by examining the flow of a client request as it is received, processed, and responded to by the server. Figure 1 illustrates this flow, where the following numbered descriptions correspond to the circled numbers in the illustration:

1. A client sends a request in the form of TCP/IP data packets that the server receives through its network interface card (NIC) or LAN on motherboard (LOM). The data packet contains TCP header information that includes packet identification and routing information as well as the actual data payload relating to the client request.

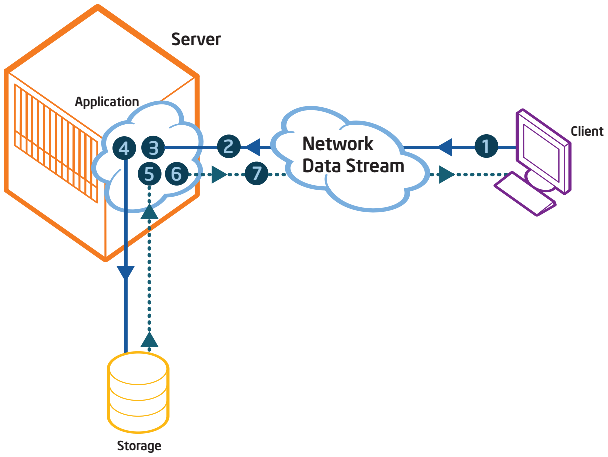


Figure 1. Data paths to and from application. Server overhead and response latency occurs throughout the request-response data path. These overheads and latencies include processing incoming request TCP/IP packets, routing packet payload data to the application, fetching stored information, and reprocessing responses into TCP/IP packets for routing back to the requesting client.

2. The server processes the TCP/IP packets and routes the payload to the designated application. This processing includes protocol computations involving the TCP/IP stack, multiple server memory accesses for packet descriptors and payload moves, and various other system overhead activities (for example, interrupt handling, buffer management, and so forth).
3. The application acknowledges the client request and recognizes that it needs data from storage to respond to the request.
4. The application accesses storage to obtain the necessary data to satisfy the client request.
5. Storage returns the requested data to the application.
6. The application completes processing of the client request using the additional data received from storage.
7. The server routes the response back through the network connection to be sent as TCP/IP packets to the client.

The above client-request/server-response cycle has remained largely unchanged for over 10 years, and exchanging data via TCP/IP continues to be the essential methodology for processing network data. In the past, when traffic volumes and speeds were low, server performance was more than adequate for the task. Today, packet traffic volumes and speeds are much higher, and they will continue to increase with the growth of network applications and appliances. The result of network data growth and faster rates of arrival translates into a direct increase in CPU overhead and a corresponding reduction in available CPU cycles for actual applications. For the end user, server application throughput is limited, data center performance decreases, and the user experience degrades.

Intel® I/OAT—A Multifaceted Solution for a Multifaceted Problem

Figure 1 illustrates the point that network I/O bottlenecks are a multifaceted, platform-wide issue. Network packets must be received, recognized, and processed in order to deliver a payload to the application. The application, while acting on a client request, often must fetch necessary data from memory and return a response to the server. And finally, the server must transform the response into TCP/IP packets and send it back to the client.

Intel I/OAT addresses network I/O bottlenecks in a three-pronged approach:

- Reducing system overhead
- Streamlining memory access
- Optimizing TCP/IP protocol computation

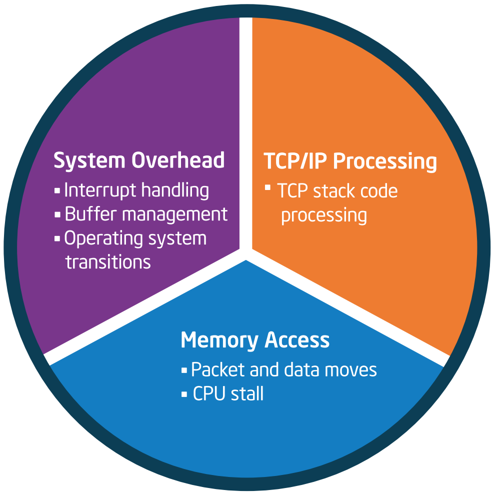
Intel I/OAT reduces system overhead through interrupt moderation, memory access moderation, parallel compute and data movement operations, and reduced context switching. Memory access and data movement efficiency is increased via pre-fetches, a direct memory access (DMA) subsystem for offloading data movement and asynchronous copies, TCP segmentation offload to the NIC or LOM, and creating affinities between data flows and specific processor cores. These techniques reduce CPU utilization for network I/O tasks, including protocol computation, and significantly reduce CPU stalls caused by cache misses and cache line bouncing.

Amidst all these developments, one primary objective is to move data more efficiently to and from applications while maintaining the integrity of the native operating system network stack. Intel I/OAT is natively supported in major operating systems and, as a result, preserves key network configurations (such as teaming and failover) while unleashing the power of Intel Xeon processors through more efficient data movement and reduced system overhead.

The Intel I/OAT Advantage

Figure 2 shows the network I/O processing segments that Intel I/OAT addresses. It is important to note that the sizes of these segments relative to each other vary depending on packet or payload sizes. TCP/IP protocol processing is constant across application I/O sizes and depends on line rate. However, due to increased system overhead at lower I/O sizes, the relative portion of TCP/IP processing varies from around 5 percent (1K I/O) to 30 percent (64K I/O) of the total CPU utilization. Similarly, memory access and system overhead vary with I/O size as a percentage of total CPU utilization.

For TCP/IP processing and memory access, it is reasonable to ask whether those issues are already covered by TCP offload engines (TOEs) and NICs enabled with remote direct memory access (RDMA). The answer is yes and no. TOE is an interface-oriented, TCP-focused approach that does not fully address the other performance bottleneck segments shown in Figure 2. As shown in Figure 3, TCP/IP processing is often a relatively small part of CPU utilization compared to system overhead and memory accesses; thus, TOE address only a small portion of the overall problem. Furthermore, given its offloading of the network stack to a fixed speed microcontroller, there is the risk that key network configurations will not function in a TOE environment.



Source: Intel Linux* analysis

Figure 2. Network I/O processing tasks.
Server network I/O processing tasks fall into three major overhead categories, each varying as a percentage of total overhead according to TCP/IP packet size.

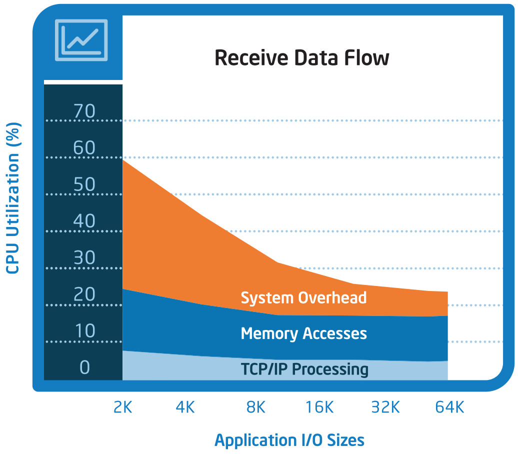


Figure 3. CPU utilization varies according to I/O size.
TCP/IP processing is fairly constant and tends to be a smaller part of CPU utilization compared to system overhead.

As for RDMA-enabled NICs (RNICs), the RDMA protocol supports direct placement of payload data into an application's memory space to reduce data movement overhead. The RDMA protocol is an addition to existing network protocols like TCP/IP and exhibits its own overhead to arrange each data transfer.

Rather than place this burden on the CPU, RDMA uses TOE as a resource to run this new workload. The combination of TOE and RDMA improves data movement and offloads the processing of two protocols. However, because RDMA requires modification of applications to make them RDMA-aware, only a few server environments can implement the RDMA protocol. Additionally, RDMA is an end-to-end protocol, so its implementation must include all systems that will transfer data. Because of these implementation issues and narrow application range, RNICs have seen limited industry acceptance.

By contrast, Intel I/OAT addresses all segments of the server I/O bottleneck problem and does it by using TCP/IP without requiring any modification of existing or future applications. The system-wide network I/O acceleration technologies applied by Intel I/OAT are shown in Figure 4 and include:

- Network Flow Affinity. Partitions the Network Stack Processing dynamically across multiple physical or logical CPUs. This allows CPU cycles to be allocated to the application for faster execution.
- Asynchronous Low-Cost Copy. Provides enhanced DMA, allowing payload data copies from the NIC buffer in system memory to the application buffer with far fewer CPU cycles, returning the saved CPU cycles to productive application workloads.
- Improved TCP/IP Protocol with Optimized TCP/IP Stack. Implements separate packet data and control paths to optimize processing of the packet header from the packet payload. This and other stack-related enhancements reduce protocol processing cycles.

The synergistic combination of these technologies in Intel I/OAT significantly increases network performance, as shown by the test results in Figure 5. The tests were conducted for both Linux* and Microsoft Windows* operating systems using two different Intel Xeon processor-based servers, one

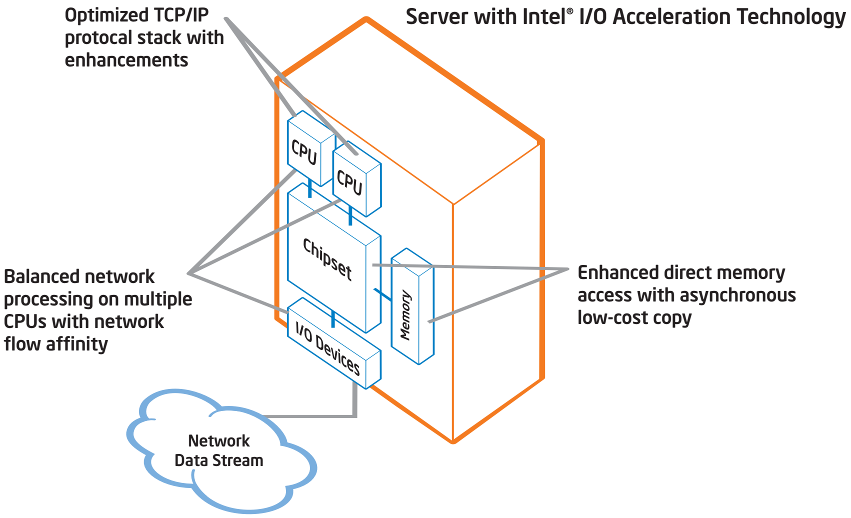

Optimized TCP/IP protocal stack with enhancements
Server with Intel® I/O Acceleration Technology
CPU
CPU
Chipset
Memory
I/O Devices
Balanced network processing on multiple CPUs with network flow affinity
Enhanced direct memory access with asynchronous low-cost copy
Network Data Stream

Figure 4. Intel® I/OAT performance enhancements. Intel® I/OAT implements server-wide performance enhancements in all major overhead categories to ensure that data gets to and from applications consistently faster and with greater reliability.

being the new Dual-Core Intel Xeon processor-based platform with Intel I/OAT. For the Linux tests at eight ports, there was over 40 percent less CPU load for the Dual-Core Intel Xeon processor with Intel I/OAT versus the platform without Intel I/OAT, and throughput was more than doubled.

Similarly, with Microsoft Windows Server 2003,* throughput with the Intel I/OAT-assisted platform was nearly doubled for eight ports. Results for CPU percent Utilization could not be obtained for the Intel® E7520-based platform because, at four ports, the processors on that platform became saturated, limiting the system's ability to report data. However, the new Dual-Core Intel Xeon processor-based platform with Intel I/OAT never saturated and only reached 70 percent CPU utilization at the full test load of eight ports.

A System-wide Solution for Higher Performance, Reliability, and Efficiency

Intel I/O Acceleration Technology moves networking I/O data more efficiently through the new Dual-Core Intel Xeon processor-based servers for fast, scalable, and reliable networking. Intel I/OAT improves network application responsiveness by unleashing the power of Intel Xeon processors through more efficient network data movement and reduced system overhead while scaling seamlessly across multiple GbE ports and maintaining the security and flexibility desired by IT managers.

For More Information

To find out more about Intel I/O Acceleration Technology, visit www.intel.com/go/ioat.

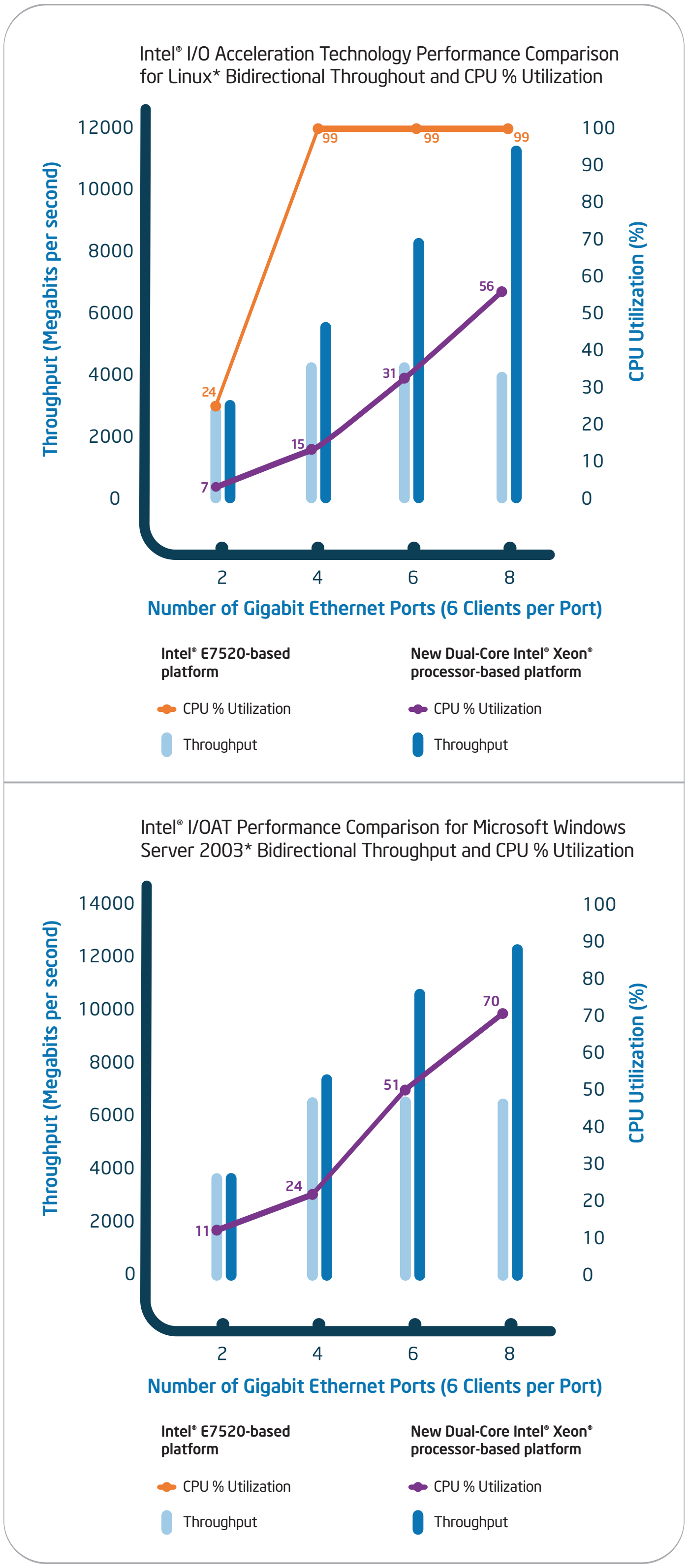


Figure 5. Network-performance comparisons for platforms with and without Intel® I/OAT. Compared to previous processors, the new Dual-Core Intel® Xeon® processor with Intel® I/OAT provides superior performance in terms of both higher throughput and reduced percentage of CPU utilization.

Printed in USA 0606/BY/PMS/PP/2.5K Order Number: 313506-001US