White Paper
Intel® Xeon® Processor E5 Family
Big Data Analytics
Cloud Computing Solutions

Big Data Technologies for Ultra-High-Speed Data Transfer and Processing

Using Technologies from Aspera and Intel to Achieve 40 Gbps WAN/LAN Data Transfer/Speeds

Executive Summary

One major challenge in high-performance cloud computing is being able to move big data in and out of the backend data center. While high-performance servers and hardware are already deployable inside the data center, and WAN bandwidth can be provisioned beyond multi-Gbps, existing transport technology cannot fully utilize the end-to-end capacity provided by the underlying hardware platform, particularly over a wide area.

Aspera develops high-speed data transfer technologies that provide speed, efficiency, and bandwidth control over any file size, transfer distance, network condition, and storage location (i.e., on-premise or cloud). Aspera's patented *fasp** transport technology is designed and implemented to have no theoretical throughput limit. Maximum transfer speeds are limited only by the available network bandwidth and the hardware resources at both ends of the transfer. Aspera's distance-neutral transfer performance means users can expect to achieve the same transfer performance over the WAN as they do over the LAN.

Aspera and Intel investigated ultra-high-speed (10 Gbps and beyond) data transfer solutions built on Aspera's *fasp* transport technology and the Intel® Xeon® processor E5-2600 product family. In Phase I of this investigation, the team learned that users of Aspera's commercially available software can achieve predictable 10 Gbps WAN transfer speeds on commodity Internet connections, on both bare metal and virtualized hardware platforms, including over networks with hundreds of milliseconds of round-trip time and several percentage points of packet loss characteristic of typical global-distance WANs. This can enable transfer speeds between 100 and 1,000 times faster than standard Transmission Control Protocol (TCP) over the same conditions. Intel Xeon processors can support 10 Gbps speeds on both bare metal and virtualized hardware platforms using Aspera's commercially available software. (Find a detailed description of the Phase I work in the Intel white paper Big Data for Life Sciences.)

While the tests were able to fully utilize a 10 Gbps Ethernet interface when running Aspera transfers in multi-threaded mode with regular packet sizes, each stream achieved a maximum throughput of about 4.2 Gbps due to the CPU and kernel overhead of interfacing with a User Datagram Protocol (UDP) socket. The processor-intensive operations contributing to this bottleneck have to do with the way in which traditional network sockets in the kernel networking stack communicate data between the transfer application and the operating system.

Phase II of the Aspera-Intel investigation focused on an experimental integration with Intel® Data Plane Development Kit (Intel® DPDK), which made it possible to directly control the network interface controller (NIC), thereby bypassing the kernel networking stack. This integration allowed Aspera to overcome the packet processing bottleneck for single-stream transfers, minimizing CPU, memory, and I/O bottlenecks.

This white paper discusses the Aspera and Intel DPDK experimental integration and the results of transfers when using this integrated software. Using the Intel DPDK capability to reduce the number of memory copies needed to send and receive a packet enabled Aspera to boost single stream data transfer speeds to 37.75 Gbps on the tested system, which represents network utilization of 39 Gbps (when Ethernet framing and IP packet headers are accounted for). The team also began a preliminary investigation of transfer performance on virtualized platforms by testing on Kernel-Based Virtual Machine (KVM) hypervisor, obtaining initial transfer speeds of about 16 Gbps. The KVM solution was not non-uniform memory access (NUMA) or memory optimized. By adding these optimizations in the future, the team expects to further extend transfer performance on virtualized platforms.

The ability to achieve such high speeds, on both bare metal and virtualized hardware systems that are truly off-the-shelf, could dramatically reduce data center hardware costs and footprint as well as the associated power and cooling costs. Users with ultra-high-speed transfer needs could significantly reduce the operating costs of transferring data into public cloud infrastructures by dramatically reducing the number of nodes involved in such transfers. With the right storage subsystem, this same inexpensive, off-the-shelf hardware system could scale wide-area transfer performance to well over 100 Gbps from disk to disk, enabling revolutionary data access times at far-flung distances and, even better, reducing both hardware footprint and costs and further lowering power and cooling costs.

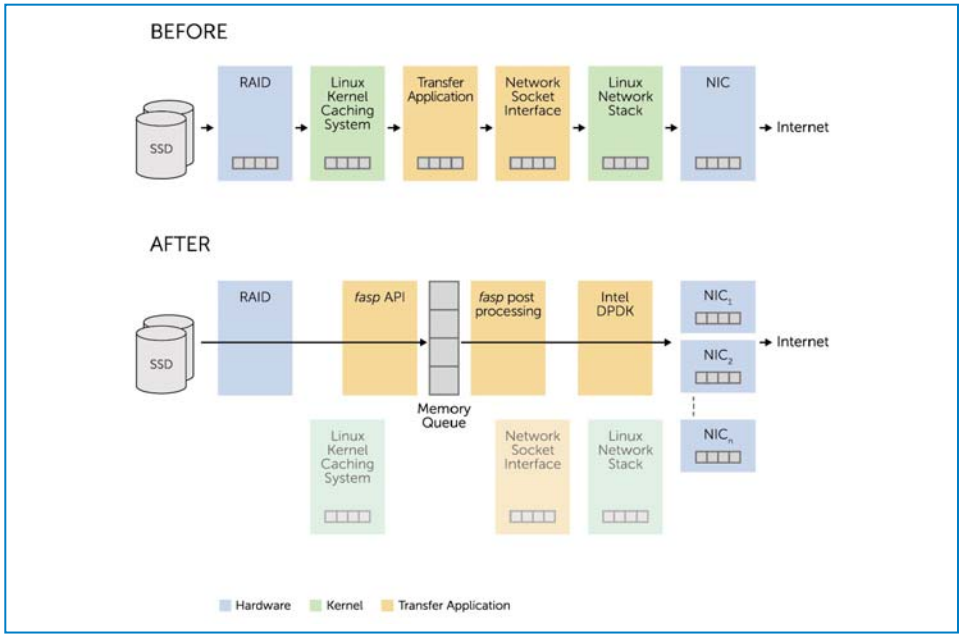

Figure 1. Data Transfer Architecture With and Without Intel DPDK Integration

Aspera-Intel DPDK Integration

In Phase II of the Aspera-Intel ultra-high-speed investigation, the goal was to explore throughput limits through direct use of Intel DPDK. Using Intel DPDK provided the ability to eliminate the traditional per-packet processing bottleneck of using a socket-based interface in the high-speed, application-level protocol. Instead, the team could concentrate on less traditional barriers in high-speed I/O such as file I/O, memory bandwidth, network loss, and basic system architecture.

Intel DPDK integration can significantly reduce the effects of the primary speed barrier for sending and receiving data (i.e., the CPU overhead in copying data through the kernel networking stack and with user space). The test team created a prototype *fasp* sender and receiver that used Intel DPDK to bypass the kernel networking stack, reading data directly from storage into designated memory that is directly accessible for transmission over the network interface. This eliminates the multiple data copies in the kernel networking stack and through user space, which conventional system architecture and traditional socket programming require.

Figure 1 shows the architectural differences between high-speed transfers that use Intel DPDK integration and the transfers that do not. As the high-level "before" diagram shows, preparing packets for transfers over the Internet involves a series of steps, each involving memory copies. Very high-speed transfers and memory- and CPU-intensive operations (e.g., at the Linux* kernel socket interface) provide a significant impediment to fully using the network bandwidth available on modern CPU architectures.

As the "after" diagram in Figure 1 shows, Intel DPDK integration makes it possible to bypass the kernel networking stack, thereby significantly reducing the number of memory copies required to send and receive data. Also, it is

important to note that in a traditional disk I/O, data moves from the application to the kernel, where it is cached. That cached data is then eventually written to disk according to OS heuristics, or copied into application memory space as appropriate. This approach makes sense when you view the disk as an I/O bottleneck and memory as being significantly faster than the disk. However, for the test, the disks approach the speed of single channel DDR3 memory. Therefore, unnecessary memory copies end up being a bottleneck to efficient I/O. While the test fully supports traditional disk I/O patterns, it also supports direct (cacheless) disk I/O, which either fully eliminates the disk-to-memory copy by copying data directly to CPU cache or uses direct memory access (DMA) to reduce disk reads to a single memory operation.

While the socket interface is the most significant barrier to high speed I/O, moving past about 10 Gbps, other factors begin to become very significant. These include:

- Memory copies
- Non-aligned memory
- NUMA locality
- Cache size
- Thread synchronization
- PCI Express* data link bandwidth

The "before" diagram in Figure 1 shows the limits imposed by traditional kernel architectures. The memory copies that Intel DPDK integration makes it possible to eliminate (e.g., network socket interface, Linux network stack, Linux disk caching) tend to require careful kernel CPU pinning to eliminate pathological behavior caused when NUMA effects are not accounted for. Using Intel DPDK dramatically simplifies this scenario. The DPDK library enabled the team to achieve a NUMA-aware application, with memory reads and writes aligned relative to the four memory controllers on an Intel Xeon processor and memory accesses that use huge pages to reduce the overhead associated with page table misses, out of the box. This approach, combined with an architecture that minimizes memory copies and CPU overhead, allowed the team to provide a very high-speed transfer architecture that works without any special kernel configuration or exotic optimizations.

Test Setup and Details

To test the performance of the Aspera and Intel DPDK integration, the team set up two identical Intel Xeon processor-based servers configured as shown in Table 1. Each of the Intel Xeon processor-based servers uses Intel DPDK and NUMA technologies to deliver superior I/O throughput. Each system is designed to be used with Red Hat Enterprise Linux* and Aspera's Intel DPDK-integrated software. The storage subsystem used XFS aligned with a hardware RAID solution to provide direct-to-CPU disk operations.

For the demonstration, the team disabled the hardware raid cache for both reading and writing and then performed direct I/O to the underlying RAID subsystem using XFS*. The team use all four 10G interfaces in a direct connect arrangement and tested single-stream transfers between the two Intel Xeon processor-based servers. The results are described in the next section.

Results of the Investigation

Using the Intel DPDK capability to reduce the number of memory copies needed to send and receive a packet enabled Aspera to boost single stream data transfer speeds to 37.75 Gbps on the tested system, which represents network utilization of 39 Gbps (when Ethernet framing and IP packet headers are accounted for). The team also began preliminary investigation of the transfer performance on virtualized platforms by testing on a kernel-based virtual machine (KVM) hypervisor and obtained initial transfer speeds of 16.1 Gbps. The KVM solution was not yet NUMA or memory optimized, and thus the team expects to obtain even faster speeds as it applies these optimizations in the future.

Table 1. Test Configuration

Hardware
▪ Intel® Xeon® processor E5-2650 v2 (eight cores at 2.6 GHz with hyperthreading)
▪ 128-GB DDR3-1333 ECC (16 x 8 GB DIMM)
▪ Twelve Intel® Solid-State Drives DC S3700 series (800GB, 6Gb/s, 2.5" MLC per server)
▪ Two Intel® Ethernet Converged Network Adapters X520-DA2 (dual port, 10G NIC with four ports total)
▪ Two Intel® Integrated RAID Modules RMS25PB080 (PCIe2 x8 with direct attach to disk)
Software
▪ DPDK 1.4 from dpdk.org
▪ Prototype Aspera *fasp** sender and receiver with Intel® DPDK integrated
▪ XFS File system
▪ 1MB RAID Stripe size for 12MB blocks

For performance analysis comparing the Aspera-Intel DPDK architecture to traditional data transfer mechanisms, the team focused on the data transmission and reception rates independent of wide-area round-trip time and packet loss to eliminate the TCP protocol bottleneck of legacy protocols and, in turn, reveal their secondary bottlenecks in sending and receiving speeds due to host-based architecture limitations. In this case, the team used Network File System (NFS) data copy and File Transfer Protocol (FTP) as the representative legacy protocols for comparison purposes and Iperf in TCP mode for a baseline measurement of legacy protocol network throughput. The relative performance of the Aspera/Intel DPDK transfers

compared to NFS and FTP is shown in Figure 2. For all tests, the team used the same servers and disk configuration, and bonded the 10g interfaces together using Linux modbonding to create a single 40g interface.

The team ran two types of tests for each technology:

- Transfers from the disk array on the remote computer to memory on the local machine (shown in blue)
- Transfers from the disk array on the remote machine to the disk array on the local machine (shown in green)

For NFS, the team used default mount options and configured the NFS server with the asynchronous async option. Iperf was configured with a 593 kB TCP window size, which corresponded to the largest window size the team could configure.

In Figure 2:

- cp represents a standard copy
- Iperf represents TCP performance
- dd shows performance using dd with 32MB block sizes to transfer a file

The tests did not indicate the best possible performance using each technology because maximizing performance requires taking into account effects such as caching, raid, memory, and CPU as well as correctly configuring the tool being used. However, the performance comparison is a good baseline from which to compare the relative performance of the transfer solution.

Please note that due to the Aspera *fasp* distance-neutral transfer performance, a complete implementation of *fasp* on top of Intel DPDK should achieve the same performance over WAN conditions as in the LAN.

Implementing the entire *fasp* transfer stack on top of Intel DPDK is beyond the scope of this phase; however, the team believes the results

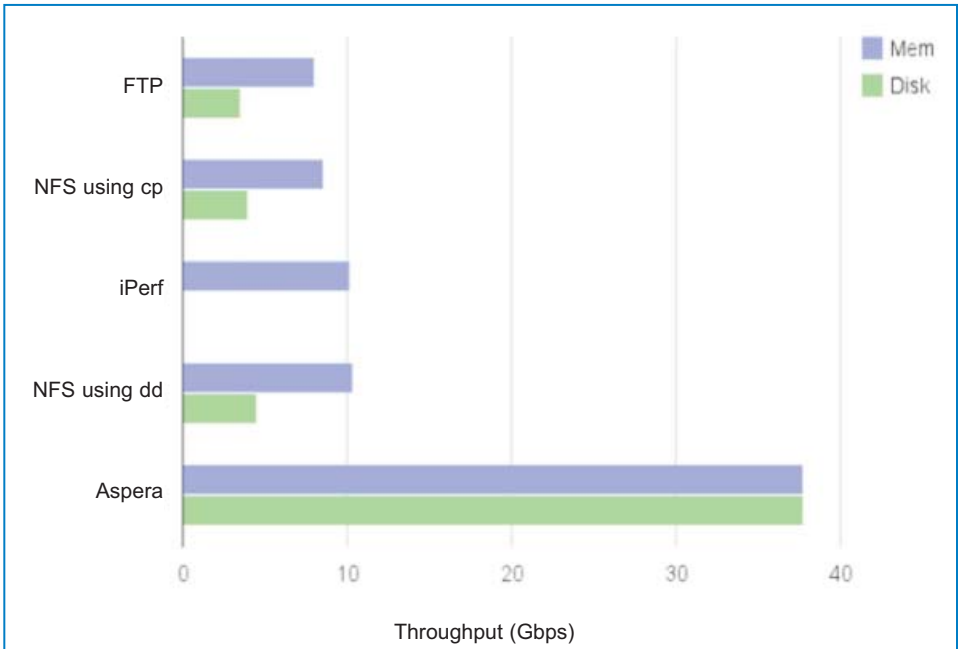

Figure 2. Relative Performance Comparison

are sufficiently promising to indicate this can be done successfully.

By contrast, the performance for the other technologies shown in Figure 2 is only for LAN transfers and will degrade according to the TCP WAN bottleneck as demonstrated in Phase 1 and multiple other studies. Unlike other high-performance solutions, the architecture is very simple, providing a nearly direct data path from disk to network card and thus eliminating the last bottleneck of per-packet processing in the host system.

Also, the solution uses commodity hardware and avoids custom drivers and other advanced system set-up. Reproducing the results requires:

- Installing the RAID cards and network adapters on the same CPU socket
- Configuring XFS in the same way as the raid adapters
- Fully disabling all hardware read and write cache

Future Performance Enhancements

A 100 Gbps Ethernet link corresponds to a transfer speed of 11.37 gigabytes per second, which is well within the PCI Express data link speed (15.75 Gbps per 16 lanes) provided by current-generation Intel Xeon processors. There remain two fundamental limitations to overcome to hit a 100 Gbps single transfer stream:

- Memory bandwidth
- Disk throughput

First, the solution must minimize unnecessary reads and writes to memory. On the Intel DPDK side, some enhancements are already a part of the roadmap such as the ability to create a network packet whose data portion points to some location in memory. This is a huge win. On the receive side, the team would like network packets to bypass main memory and go directly into the L3 cache of the receiving processor. This

may already be occurring to some extent through the use of data direct I/O (DDIO), but additional metrics would be helpful to effectively tie a specific core to a network interface. Both of these changes could reduce memory bandwidth utilization by roughly half for a single read and write to memory per datagram.

Second, the solution must address disk performance. The ideal disk subsystem would be a no-copy software RAID implementation aligned around the 4 KB block size of the Intel® Solid-State Drive DC S3700 series. For instance, with 16 solid-state drives, it should be possible to read a 64KB block in the same amount of time it would take a single disk to read a 4KB block. In practice, one would want to read data directly to the L3 cache, making it possible to read, process, and queue data for network transmission without using main memory.

Conclusions

Phase I of the Aspera-Intel ultra-high-speed investigation showed it is possible to use Aspera's commercially available software running on off-the-shelf Intel® hardware with built-in support for DDIO and single-root I/O virtualization (SR-IOV) to achieve predictable 10 Gbps transfer speeds over global WAN conditions on both physical and virtual platforms using standard packet sizes. Intel Xeon processors support the DDIO and SRI-OV optimizations used, enabling 10 Gbps transfer speeds on both bare metal and virtualized hardware platforms using Aspera's commercially available software.

In Phase II, the team provided a modern approach to network I/O that is compatible with high-throughput, low-latency disk I/O, virtualized machines, and high-performance network architectures. This was accomplished with Intel DPDK, which offers high-performance I/O routines optimized to take advantage of NUMA, Intel® Ethernet Adapters, and multi-channel memory controllers. The API making it possible to use this interface is similar to the existing *fasp* API, which has been successfully used in complex, high-performance vendor workflows.

The team tested transfers on the Aspera-Intel DPDK integrated software with a target transfer rate of 40 Gbps. Using the Intel DPDK capability to reduce the number of memory copies needed to send and receive a packet enabled Aspera to boost single stream data transfer speeds to 37.75 Gbps on the tested system, which represents network utilization of 39 Gbps (when Ethernet framing and IP packet headers are accounted for). The team also began preliminary investigation of transfer performance on virtualized platforms by testing on KVM hypervisor, obtaining initial transfer speeds of 16.1 Gbps, which can be further extended in the future with additional NUMA and memory optimizations.

The ability to achieve such high speeds—on both bare metal and virtualized hardware systems that are low-cost and off-the-shelf—could dramatically reduce data center hardware cost and footprint and the associated power and cooling costs. Furthermore, users with ultra-high-speed transfer needs could significantly reduce the operating costs of transferring data into public cloud infrastructures by dramatically reducing the number of nodes involved in such transfers. The test team believes that with the right storage subsystem, this same architectural approach would scale transfer performance to more than 100 Gbps over the WAN, from disk to disk, thereby enabling revolutionary data access times at far-flung distances and even better hardware footprint and cost reductions, as well as lower power and cooling costs.

Learn more about the Intel Xeon processor E5 family **here**.

Learn more about Aspera *fasp* at **www.asperasoft.com/technology/transport/fasp**.

Intel DPDK: Packet Processing on Intel® Architecture

With Intel® processors, it's possible to transition from using discrete architectures per major workload (application, control, packet, and signal processing) to a single architecture that consolidates the workloads into a more scalable and simplified solution. As a result, developers may be able to eliminate special-purpose hardware such as network processors (NPUs), co-processors, application specific integrated circuits (ASICs), and field-programmable gate arrays (FPGAs).

This is possible, in large part, due to the Intel® Data Plane Development Kit (Intel® DPDK), a set of software libraries that can improve packet processing performance by up to 10 times. As a result, it's possible to achieve over 80 Mpps throughput on a single Intel® Xeon® processor and double that with a dual-processor configuration.

To learn more, visit **www.intel.com/go/dpdk**

1113 KK/SS