White Paper

Network Hardware Optimization
Kernel Bypass Solutions

GateSpeed Software Stack Maximizes Performance Across Intel Ecosystem

Tests conducted at Intel Labs demonstrate a scalable, software-only solution for ultra-high-throughput network requirements. With per-core performance above 200Gbps[1], the GateSpeed stack offers network architects a plausible alternative to approaches like using ASICs for designing next-gen networks using Intel® architecture devices.

Since the introduction of multi-core, multi-threaded CPUs, acceleration tools like Data Plane Development Kit (DPDK), Vector Packet Processor (VPP) and network hardware virtualization such as Single Root Input/Output Virtualization (SR-IOV) have emerged to help network operators (NetOps) address rising data volume and performance requirements for increasingly complex applications. These complex applications are being deployed across diverse modes, spanning bare metal, virtualized, and containerized environments, all which demand network optimization.

As network architects and operators face complex challenges in a growing digital economy, software-based networking solutions that maximize scalability and efficiency offer a unique competitive edge.

GateSpeed's proprietary software stack and application platform equip engineers with the tools needed to successfully design and deploy high-throughput networks by maximizing per-core performance.

In recent lab tests, the GateSpeed stack demonstrated performance thresholds up to 200Gbps throughput on a single CPU core. With terabit-per-second throughput requirements on the horizon, the GateSpeed stack's linear scalability provides a foundation for tomorrow's network requirements.

Evolution of Kernel Bypass

Broadly speaking, kernel bypass was primarily designed to overcome lower network layer data flow bottlenecks within Linux. The leading example of kernel bypass solutions is DPDK, an Intel designed, open source set of data plane libraries and network interface controller polling-mode drivers for offloading TCP packet processing from the operating system kernel to processes running in user space.

There is no doubt that DPDK's introduction opened a significant new pathway for network developers to access hardware capabilities to their fullest. It laid a robust foundation for developers worldwide to build on. And while DPDK gained in popularity it required complex integration to achieve enterprise scale.

The need for a robust and scalable kernel bypass platform that simplifies, and speeds up open source software remains the holy grail for NetOps teams and their most important customers - DevOps teams.

GateSpeed's stack offers engineering teams a robust and composite networking stack that allows seamless integration of applications (see NGINX test results below) operating on Intel® architecture hardware.

Table of Contents

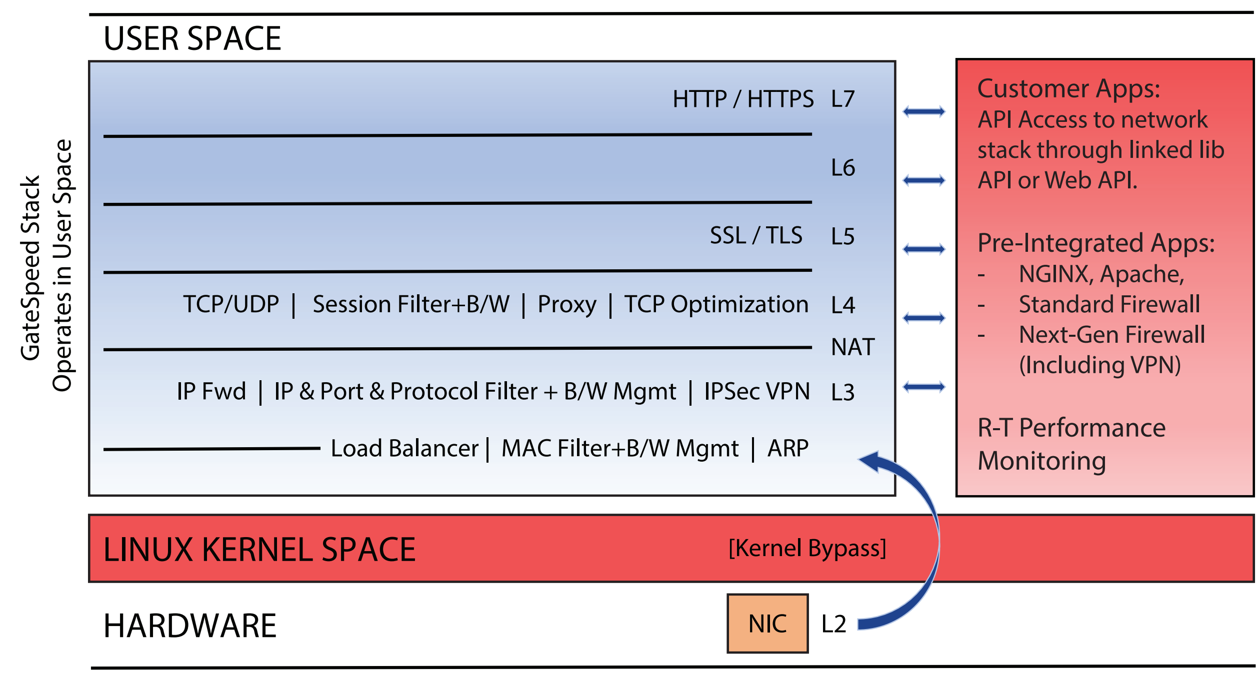


Figure 1. GateSpeed stack.

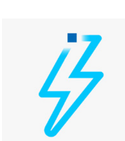
The GateSpeed Software Stack: Maximizing Kernel Bypass Impact

GateSpeed's proprietary stack (see Figure 1) helps app developers bypass the Linux kernel. The GateSpeed platform provides a performance foundation for applications including high-throughput router, web server acceleration, VPN, firewall and applications that rely on high throughput to maximize user experience and user density.

Additional GateSpeed features include Express fast core, scheduler, forwarding engine, traffic management, TCP session/state machine, proxy, and intelligent VLN congestion control switcher.

GateSpeed: Multi-core Scalability and Efficiency

The GateSpeed stack eliminates the bottleneck between network interfaces and applications. It combines scalability and efficiency and seamlessly integrates with customer-defined architectures.



Efficiency: GateSpeed is CPU-based and performance competitive with ASIC solutions at less cost and power consumption and greater operational flexibility. For example, it can perform layer-3 forwarding of packets at 200 Gbps with a single Intel® Xeon® Scalable CPU core utilizing only 20% of the processing power[1] – leaving a significant amount of processing headroom for scaling throughput. In recent field testing described later in this paper, this per-core efficiency translated to up to an 8:1 reduction in RAM buffer requirements and greater performance for applications for certain scenarios.

Scalability: GateSpeed's stack can scale linearly by adding more CPU cores, and also operates across multiple CPU types – ranging from low-cost Intel® Atom® CPUs to server-class Intel® Xeon® Scalable CPUs. Combined with its highly efficient core utilization, GateSpeed's scaling abilities allows customers to be prepared for both the surge in network speeds driven by forthcoming terabit Ethernet standards or extracting an order of magnitude better performance from legacy hardware.

Enterprise-Ready Applications: Experienced NetOps professionals can attest to the laborious path they have to traverse in fully utilizing DPDK. The open source package does not come with various kernel capabilities that users of Linux take for granted. In response, GateSpeed has designed layer 2 through layer 5 support to make DPDK's benefits fully realizable. By providing ARP, NAT, VPN and session-level bandwidth management to an optimized TCP stack with SSL support, GateSpeed can support both NetOps duties and DevOps needs.

Performance Test 1: Site-to-Site IPsec VPN

For the first GateSpeed performance test, the company simulated a site-to-site IPsec VPN tunneling application (see Figure 2), to assess the scalability and efficiency of the GateSpeed stack leveraging DPDK to maximize IPsec VPN throughput.

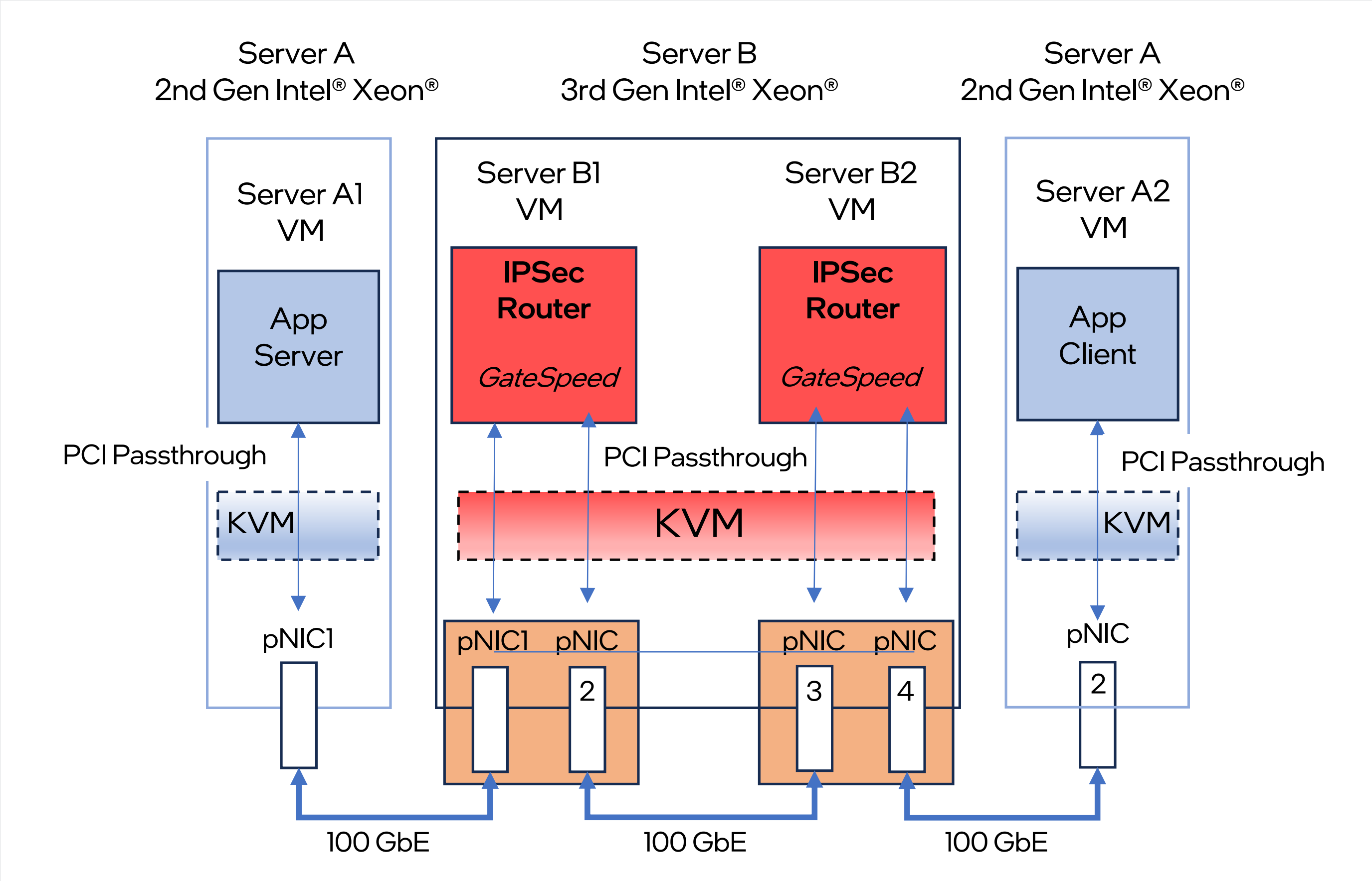


Figure 2. Site-to-Site IPsec VPN.

Tests were conducted both with and without Intel® Quick Assist Technology (Intel® QAT) to showcase enhanced cryptographic function handling with hardware assistance. Throughput was measured using common network packet sizes ranging from 64 bytes to 1455 bytes. IPsec encryption utilized AES-128-CBC-HMAC-SHA1 which is a combination of AES-128 in cipher block-chaining mode and HMAC-SHA1 hash-based message authentication code.

Server A hosted virtual machines (VM) A1 and A2 and is based on an Intel® Xeon® Gold 6230N processor-based server with two 100 GbE Intel® E810-DQCA2 network interface cards (NICs). Server B hosted virtual machines B1 and B2 and was powered by an Intel® Xeon® Gold 6342 processor and featured two Intel® Ethernet Network Adapter E810-2CQDA2 operating in bifurcated mode, effectively providing four 100 GbE NICs. The Intel® QAT used on server B was provided by an Intel®, 8970 PCI adapter.

All NICs depicted in the illustration were physical NICs. SR-IOV was employed in PCI passthrough mode to eliminate data bottlenecks. KVM, being a type 1 hypervisor, enabled VM processes to execute directly on the host CPU, minimizing virtualization overhead.

The test results (figures 3 and 4) showed GateSpeed achieved near line-rate utilization for 1455-byte packets with six CPU cores using software-based encryption. By utilizing two Intel® QAT instances on the Intel® 8970 adapter replacing two of the CPU cores, it achieved line-rate utilization while reducing the total core count needed for the application to four. The remaining two cores used in this configuration utilized software-based encryption.

Figure 3 shows the results of the tests that utilized only software-based encryption with the GateSpeed-enabled IPsec VPN application delivering 100Gbps throughput on six CPU cores.

Figure 4 IPsec VPN results showcase the capability to amplify performance gains through Intel® QAT acceleration. VPN networks can achieve a throughput of 100Gbps by harnessing the power of four cores: two cores leverage the AESNI_MB library for software-based encryption, while the other two cores harness an Intel® QAT card for hardware-based encryption.

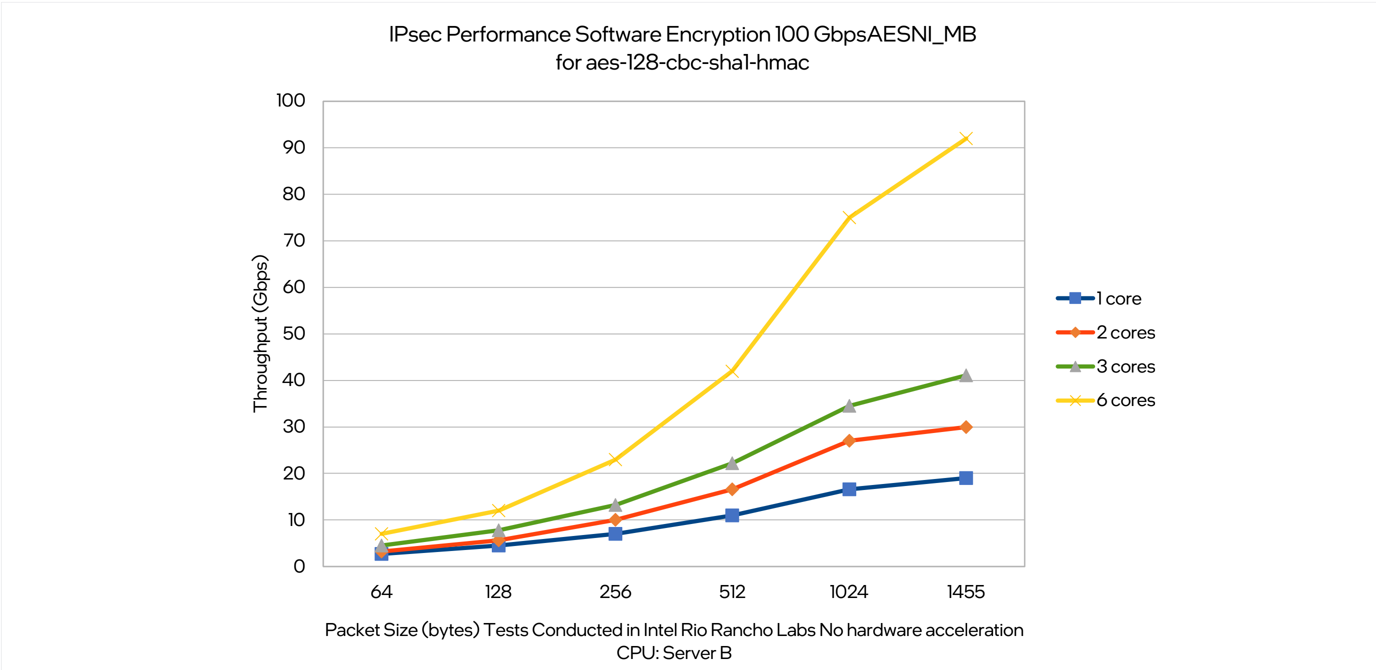


Figure 3. Six-core 100Gbps VPN results.

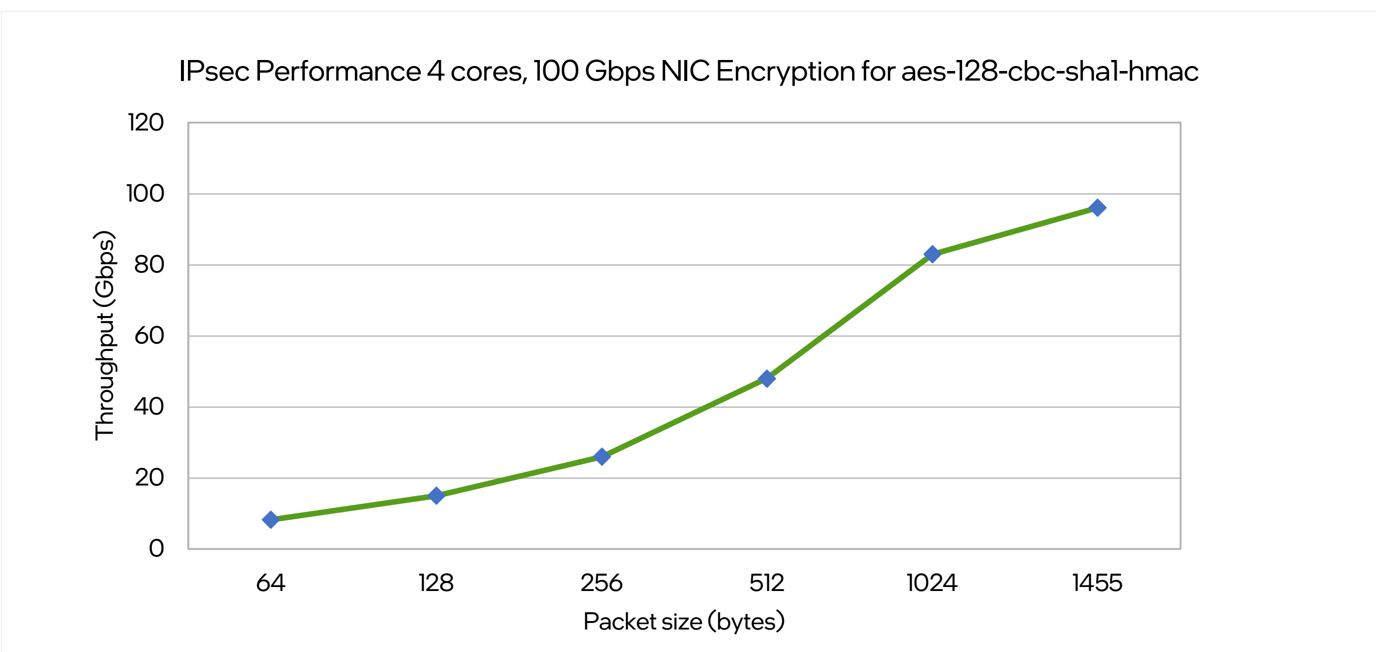


Figure 4. 100Gbps VPN on four cores w/ Intel® QAT.

Performance Test 2: NGINX Web Server Acceleration

In this assessment, GateSpeed sought to evaluate the performance, scalability, and adaptability of the GateSpeed stack when used to support network applications. NGINX, a widely deployed web-server, was used as the test application. The evaluation involved simulating client-to-web server traffic, encompassing both HTTP and HTTPS protocols, while varying packet sizes and concurrent client connections. SSL encryption was applied using both software-based and Intel® QAT hardware-based encryption methods. Key metrics, including throughput, core utilization, and core usage count, were collected (see Tables 1 and 2).

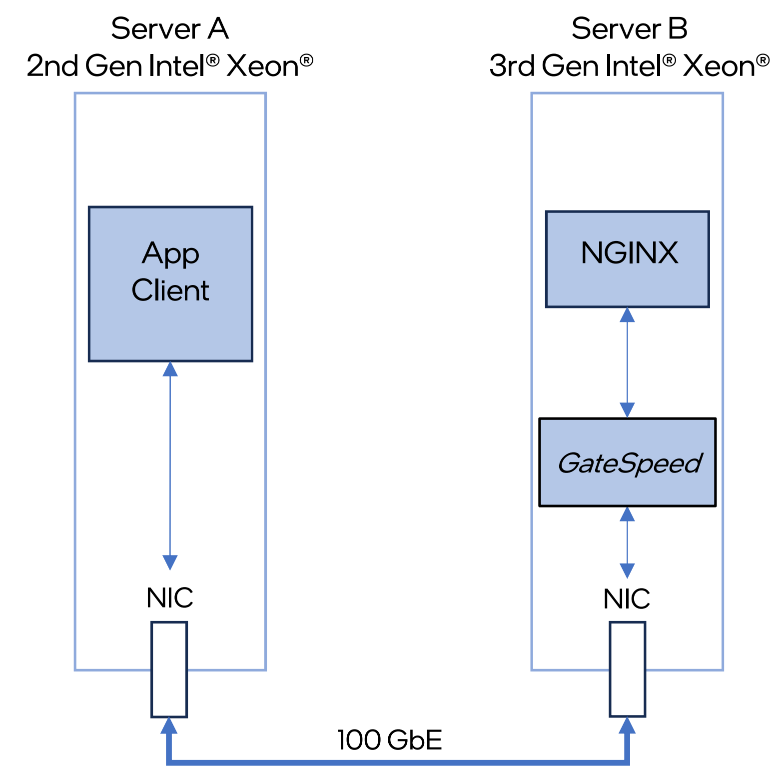


Figure 5. Server configuration for NGINX web server acceleration test.

Server A hosted the client application and featured an Intel® Xeon® Gold 6230N processor with two Intel® Ethernet Network Adapters E810-CQDA2. Server B, hosting NGINX, utilized an Intel® Xeon® Gold 6342 processor and two Intel® Ethernet Network Adapters E810-2CQDA2.

The results of the tests demonstrated that GateSpeed effectively scaled the NGINX deployment to achieve near line-rate data processing at 100 Gbps, utilizing only four or fewer processor cores for HTTP traffic. Expecting greater CPU utilization for HTTPS, leveraging Intel® QAT, testing revealed a significantly better than stock Linux kernel performance and with room for further optimization which would realize the potential to get much closer to the HTTP numbers (see Table 1). For network-centric applications that must service high data rates, the tests illustrate the performance enhancements that GateSpeed stack brings. It can support high user densities without compromising user experience and taxing hardware costs.

Test case	Concurrent Connections	File Size	NIC Throughput	CPU Utilization* (Remaining cores idle)
1	5,000	2MB	85Gbps	2 cores - 95% utilization each
2	10,000	2MB	82Gbps	2 cores - 96% utilization each
3	20,000	2MB	82Gbps	2 cores - 95% utilization each
4	40,000	2MB	82Gbps	2 cores - 95% utilization each
5	5,000	10KB	90Gbps	4 cores - 95% utilization each
6	10,000	10KB	88Gbps	4 cores - 95% utilization each
7	20,000	10KB	88Gbps	4 cores - 95% utilization each
8	40,000	10KB	82 Gbps	4 cores - 95% utilization each

Table 1. HTTP performance results.

Note: With minimal tuning we expect throughput to increase to above 90 Gbps for HTTP and mid-70 Gbps to 80 Gbps range for HTTPS.

Test case	Concurrent Connections	File Size	NIC Throughput	CPU Utilization* (Remaining cores idle)
1	5000	2MB	74Gbps	4 cores - 95% utilization each
2	10,000	2MB	68Gbps	4 cores - 95% utilization each
3	20,000	2MB	66Gbps	4 cores - 95% utilization each
4	40,000	2MB	64Gbps	4 cores - 95% utilization each
5	5000	10KB	71Gbps	4 cores - 95% utilization each
6	10,000	10KB	67Gbps	4 cores - 95% utilization each
7	20,000	10KB	63Gbps	4 cores - 95% utilization each
8	40,000	10KB	61Gbps	4 cores - 95% utilization each

Table 2. HTTPS performance results. Utilized Intel® QAT hardware assist for SSL encryption.

GateSpeed + Intel: Simplifying Hardware + Software Acceleration

Leveraging GateSpeed's firsthand experience with open-source software, the company has developed a network software stack that is straight forward to integrate with DPDK and is QA-tested for reliability and robustness.

GateSpeed's approach builds on the functionality and beneficial economics of open source DPDK, yet is easier to integrate into a server, freeing DevOps teams to focus on their strengths. As seen in the test results in this paper, the GateSpeed stack delivers kernel bypass functionality with improvements in CPU utilization and performance. It allows data to be moved at line speed, fully maximizing hardware capabilities.

(1) Linking via Standard GNU Tools: Integrating this stack with customer applications is as simple as compiling and linking GateSpeed libraries via standard GNU tools. Customer applications interact with the GateSpeed stack through network socket functions for communication and a choice of programmatic or RESTful web-based APIs for management. To scale performance, the same base binaries are duplicated across CPU cores.

(2) Dynamically Linked Socket Library: In situations where application source code is unavailable the approach to integration is to override the existing Linux socket library and replace it with the GateSpeed socket library. This method is reminiscent of mainframe patching. In fact, it is the same approach. GateSpeed's dynamically linked socket library is loaded to the same location where the application accesses the standard Linux socket library whether through dynamically linked or dynamically shared libraries.

(3) Containerized and Virtualized Solutions: GateSpeed has also developed SDK, API and standard virtualization approaches, allowing for turnkey, automated implementations of our acceleration tools in next-generation, cloud-native operating environments.

GateSpeed Optimizes Performance Across Market Segments and Full Intel Ecosystem

GateSpeed delivers performance needed by applications in market segments where user experience and user density must be simultaneously maximized. These two KPIs are increasingly important to emerging multi-cloud fabric applications, for example CDNs and streaming services offered as real-time SaaS. In this application, the fabric isn't just looking to optimize application performance but also enable the lower network layers to leverage the hardware's full potential towards their KPIs. Kernel bypass methods have never been more relevant to growth. GateSpeed's multi-layer optimization suite gives this segment a complete alternative to not just standalone DPDK but also legacy, and difficult to maintain, custom development that have diverged from standards.

While CDN and streaming services are the here and now segments of interest, GateSpeed's capabilities are also highly relevant to private and public 5G network hardware and software companies. In these networks, GateSpeed's stack is directly addressing the challenge of edge computing among other applications. The edge, MEC or even private networks, are space-constrained and need to process as much data as fast as possible locally to lower latency. The challenge is two-fold: uncompromising user experience and maximizing user density. GateSpeed's efficiency and scaling when combined with a ready-to-go stack offers this segment an easy to integrate solution to accelerate growth.

Last but not least, AI applications also need the performance delivered by GateSpeed. Today the infrastructure to develop large language models (LLMs) and other supporting models is capital intensive and highly specialized. GateSpeed is working to democratize compute access. Kernel bypass is key to the training and broader functioning of the AI infrastructure as the demands for faster training - high speeds of data consumption and transfer, are growing. GateSpeed's current solution and roadmap are attuned to Intel's work in this space and will provide Intel customers with a viable ramp to leverage its compute power to its fullest potential.

Learn More

GateSpeed Technology

Intel® Network Builders

Data Plane Development Kit

Intel® Xeon® Scalable Processors

Intel® 800 Series Ethernet Adaptors

Intel® QuickAssist Technology

Server A	Server B
Zs09): 2-node, 2x 2.3 GHz, 20-core Intel® Xeon® Gold 6230N CPUs (Zs09) with 384 GB (24 slots/ 16GB/ 2666) total DDR4 memory, microcode 0x5002f01, HT on, Turbo off, Network adapters: 2x 100 GbE, E810-2CQDA2, storage: 512GB SSD. OS: Ubuntu 20.04.1, kernel: 5.8.0-43-generic, compiler: Ubuntu 9.3.0-17ubuntu1~20.04. Other software: DPDK (22.07), Intel® QAT (QAT.L.4.20.0-00001), Nginx (1.18.0-0ubuntu1.4), Pktgen (22.04.1), Seastar (20.05).	(Zi06): 2-node, 2x 2.8 GHz, 24-core Intel® Xeon® Gold 6342 CPUs (Zi06) with 512 GB (32 slots/ 32GB/ 3200) total DDR4 memory, microcode 0xd000375, HT on, Turbo off, Network adapters: 2x 100 GbE, E810-2CQDA2 and 1x XXV710 25 GbE, storage: 960GB SSD Other hardware: Intel® QuickAssist Adapter 8970. OS: Ubuntu 20.04.2, kernel: 5.15.0-46-generic, compiler: Ubuntu 9.4.0-1ubuntu1~20.04.1. Other software: DPDK (22.07), Intel® QAT (QAT.L.4.20.0-00001), Nginx (1.18.0-0ubuntu1.4), Pktgen (22.04.1), Seastar

Table 3. SUT Specifications.

intel®

Notices & Disclaimers

[1] See Table 3 for system under test (SUT) configuration.

Performance varies by use, configuration and other factors. Learn more on the Performance Index site.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

0124/LV/H09/PDF ♻ Please Recycle 358310-001US