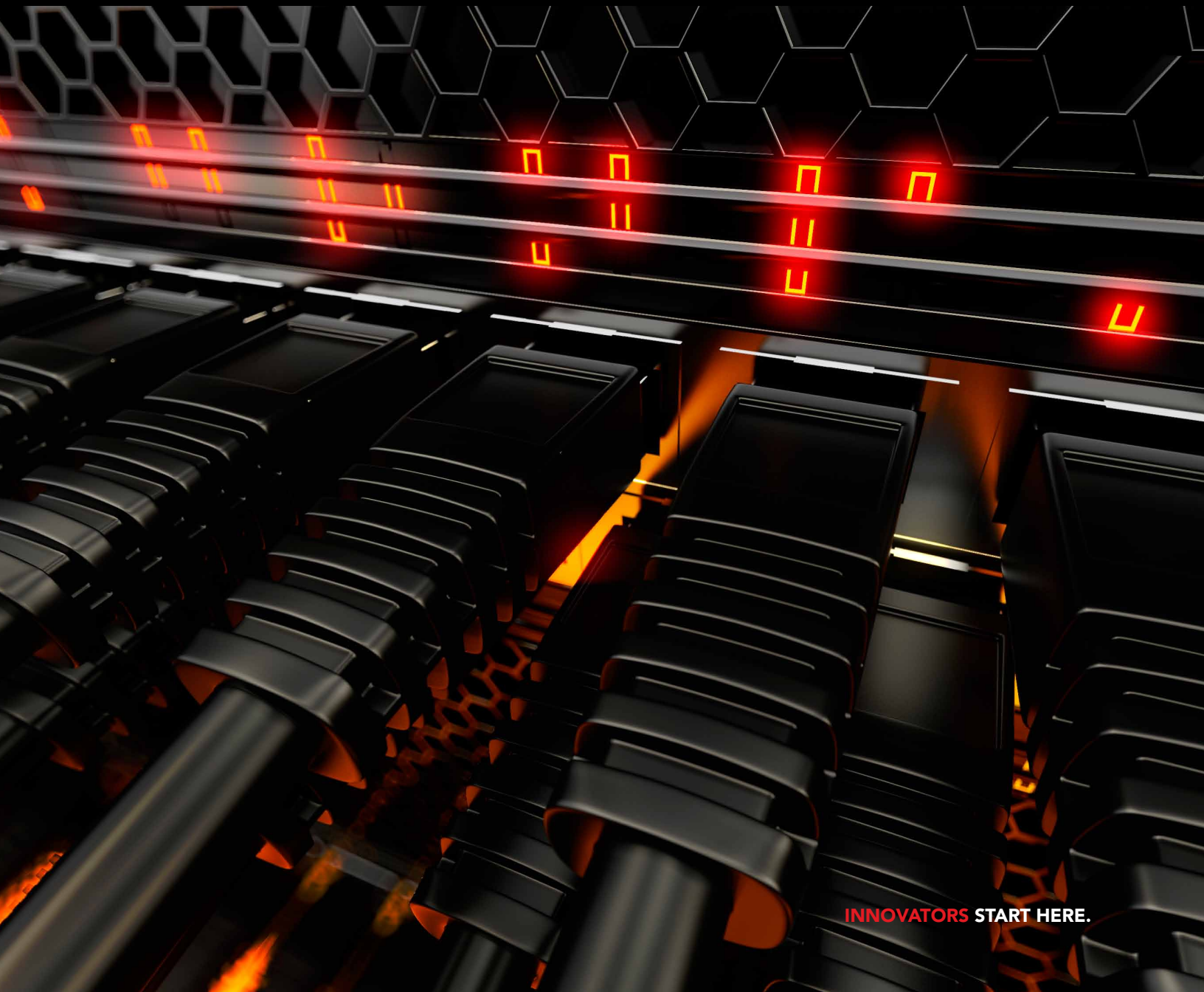
WIND RIVER

ACCELERATED DEEP PACKET INSPECTION FOR NETWORK SECURITY APPLICATIONS

Delivering High-Performance DPI on Intel Xeon Processor with Wind River Content Inspection Engine



EXECUTIVE SUMMARY

The requirement for advanced security equipment to scan data streams for malicious content at line rate presents significant challenges to network security vendors. Security equipment overwhelmed by the data rates of present-day networks is more likely to miss attacks, leading to increased risks of security breaches. While high-speed content scanning, also known as *deep packet inspection* (DPI), exists today, this technology typically requires purpose-built hardware to operate at high speeds, a solution that often leads to increases in development and manufacturing costs.

A software-based approach that takes advantage of multi-core Intel® architecture can provide a cost-effective and scalable solution that also has the flexibility to evolve with the changing needs of the system.

Wind River® Content Inspection Engine is a software pattern-matching solution designed for both single-core and multi-core processors. It offers a software-based DPI solution that can scale from under 1Gbps to 160Gbps, depending on the number of cores used. By providing DPI technology optimized for multi-core processors, it delivers a cost-effective software solution for scanning data content at line rate in security equipment ranging from small network appliances to large network elements.

TABLE OF CONTENTS

DEEP PACKET INSPECTION

Newer processor architectures with higher clock rates, larger caches, and other advances have enabled network security equipment to incorporate capabilities previously found only in end systems. No longer limited to just bridging and forwarding, network elements are now scanning packets as they pass through. Taking advantage of this technology, network administrators now deploy intrusion detection/prevention systems (IDS/IPS) and network antivirus and malware scanners alongside the traditional firewall. In some cases, several security functions are incorporated into a universal threat management (UTM) appliance.

Unlike the traditional firewall that looks at packet headers only, this more advanced type of equipment uses DPI technology to examine the content of each packet to detect threats. Instead of basing security decisions solely on fields in the packet header, DPI technology allows a security application to peer deep into the contents of a data stream to try to identify harmful intent. This more detailed examination of the data stream has an added cost, however, because scanning the content of data streams is CPU intensive and can become unfeasible as the traffic load increases.

The efficacy of DPI as a tool, therefore, depends largely on how well it performs under load. A system that performs flawlessly under contrived theoretical conditions but fails under heavy traffic presents a major security problem. Consider the analogy of a country's border services agent who is given the new mandate to pull over every automobile to conduct a detailed search of its contents. While this approach may provide added security at remote crossings where traffic is sparse, it would cause severe congestion at major crossings and result in chaos as border agents arbitrarily turn back automobiles to relieve congestion, or worse, to blindly allow entry without any controls. This is exactly what can happen in security equipment that cannot keep up with the incoming data rates. Packets can drop off queues, causing end systems to retransmit, further exacerbating the situation, or packets can be admitted blindly in a fail-open condition.

To prevent this from occurring and to increase performance, network security vendors typically relied on hardware-assisted DPI technology to perform detailed scanning at network speeds. This purpose-built silicon was expensive and difficult to use and often imposed a different programming paradigm on the software, making it frustrating to maintain and costly to deploy across the product line.

The availability of multi-core processors, however, has created the opportunity for well-designed DPI software to approach and even exceed previous hardware-based solutions in performance.

PATTERN MATCHING

Central to many DPI implementations is the notion of pattern matching, the ability to compare and match incoming byte streams against a database of known offending patterns called *signatures*. These signatures represent potential malicious content and can take the form of simple literal strings or more complex patterns such as a specific arrangement of bytes broken up by a variable amount of other possibly irrelevant data. This latter type of signature is often described using some form of regular expression syntax, sometimes combined with proprietary grammar.

While literal searches can themselves be quite intensive, regular expression searches require significant CPU resources that can become problematic to perform at high speeds, especially when searching for thousands of signatures. Figure 1 shows the marked differences in performance of a representative system running a basic firewall performing simple packet filtering and the same system running deep packet inspection against a set of rules and patterns.

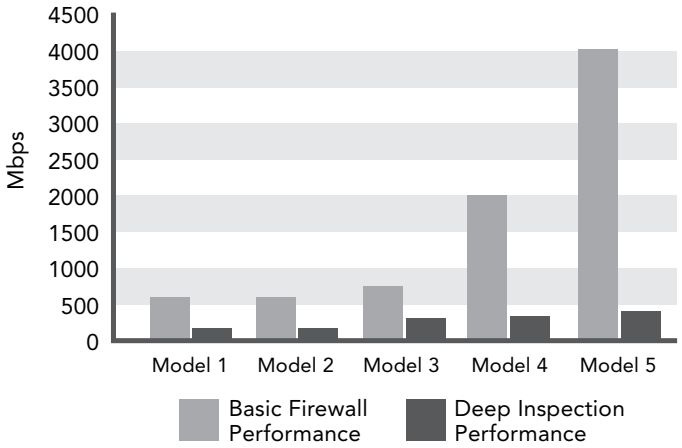


Figure 1: Firewall performance—basic vs. deep inspection

PACKET PROCESSING ON INTEL ARCHITECTURE PLATFORMS

There are a number of reasons why software running on Intel architecture platforms can achieve superior packet processing performance. Intel's tick-tock strategy of rapid introduction of new micro-architectures coupled with improvements in process technology has enabled IA multi-core processors to extend their lead in both control and data plane processing.

Recent innovations include the replacement of the front-side bus (FSB) with the point-to-point Intel QuickPath Interconnect, symmetric multi-threading, support for non-uniform memory access (NUMA), embedded and wider memory controllers, as well as new streaming instructions for even faster cyclic redundancy check (CRC) calculations. This tighter integration directly results in packets being brought in from the NIC and deposited into local memory more efficiently than ever before.

Once the packets are in memory, application software can use Intel Data Plane Development Kit (DPDK) and Intel QuickAssist Technology to speed up packet processing with features such as interrupt-free packet reception and transmission, pre-fetching and cache warming, NUMA awareness, real-time buffer management and zero-copy buffers, lockless rings, and IA-optimized longest-prefix match and flow classification.

These features, along with the high clock rates and large caches in IA processors, make the IA multi-core platform well suited for both low-touch and high-touch applications, enabling market-leading packet processing capabilities as well as accelerating intensive data path workloads such as cryptography, compression, and deep packet inspection.

Backed by a strong set of supporting tools, these architecture advancements together with Intel DPDK and Intel QuickAssist Technology are enabling a new generation of products that can deliver the most scalable and best performing solutions quickly and efficiently to the market.

A good pattern matcher should be able to perform comparisons of the incoming byte stream against the signature database with deterministic performance regardless of the number of signatures. In other words, the pattern matcher should perform much better than the brute force method of comparing the incoming byte stream with each signature sequentially. Going back and forth repeatedly in a data stream is extremely cache-inefficient and does not scale as the number of patterns grows.

Some software-based pattern matchers overcome this brute force approach by using trie-based algorithms, where the set of patterns being searched is organized into data structures that map the relationship among patterns. When an incoming packet arrives, the software merely walks the map to look for matches, similar to how some IP address lookups are performed.

While this represents an improvement over brute force approaches as the number of patterns grows, it also has its deficiencies. Depending on the number of signatures and the size of the processor cache, walking a trie in this way may require quite a few memory accesses per packet, with the worst case being one memory access per byte received, in which case performance may not turn out to be an improvement over the brute force method it is trying to replace.

Therefore, while DPI technology is becoming increasingly sophisticated, it is also creating the challenge of effectively scaling performance. The deeper and more granular the inspection, the more processing is needed; and the more important it is to find alternatives to current published state-of-the-art mechanisms for scanning.

WIND RIVER SOLUTION

While many pattern matchers in the industry are implemented with simple sequential approaches, with cache-unfriendly algorithms, or as pieces of specialized hardware that are often challenged by latency and scalability issues, equipment manufacturers can now take advantage of the Wind River software pattern-matching solution to cost-effectively drive and scale DPI performance.

Wind River Content Inspection Engine is a portable, OS-independent, multi-threaded software pattern-matching library. Easy to integrate, Content Inspection Engine is a drop-in replacement for libPCRE that not only supports a large subset of Perl Compatible Regular Expression (PCRE) syntax but also provides much better performance than libPCRE. When deployed on an Intel architecture platform, Content Inspection Engine takes advantage of features such as hyper-threading, receive side scaling, and SIMD instructions to provide up to 160Gbps in scanning performance. Aside from classic regular expressions, Content Inspection Engine supports a wide variety of other signatures required for most security and data networking applications, including anchors, character classes, and bounded repeats.

Unlike sequential approaches that go back and forth over the data stream repeatedly for each pattern in the database, Content Inspection Engine performance is not directly dependent on the number of patterns being searched. The data stream is scanned for all regular expressions in the signature set simultaneously, and matches are returned to the application as they are found. Content Inspection Engine performance is deterministic and does not exhibit drastic swings in performance usually found in traditional scanning systems.

Content Inspection Engine is capable of scanning each incoming packet independently or as a recombined data stream to detect attacks that span packets. A security application, for example, can reassemble a Transmission Control Protocol (TCP) stream and invoke the Content Inspection Engine library for scanning. Content Inspection Engine keeps track of any partial matches so that it can restart scanning from where it left off when more data arrives on that stream.

Small Signature Footprint

A key component of any regular expression-searching solution is the compiler, which reduces the set of signatures to byte code that can be understood by the underlying pattern-matching engine. Some compilers produce a small signature footprint, but these tend to be for engines that support sequential searches where the benefit of a small database is offset by poor scalability.

Other compilers build large and complex databases, which contain pattern relationship information to allow searches of thousands of signatures to be executed in parallel, but at the expense of a large memory footprint. A database that allows parallel scanning is inherently larger because it must store the relationships among all the patterns, which could lead to exponential growth depending on the nature of the signatures. The downside is that a large database can overflow from processor cache, leading to excessive memory accesses, sometimes as often as one access per byte processed.

While both of these approaches may produce good results in contrived scenarios, they may lead to suboptimal performance when running in a real-world environment with thousands of signatures and thousands of data streams. Content Inspection Engine supports parallel scanning for thousands of signatures without the associated large memory footprint inherent to traditional parallel scanning databases. Using proprietary techniques, the Content Inspection Engine compiler builds a database that separates out the key portions of the signature set, resulting in a database footprint small enough to fit entirely into a processor cache for most use cases. By keeping the database compact, external memory accesses are rarely required under normal operation. Depending on configuration, an external memory access often only takes place when a match occurs, which is the ideal behavior for a pattern-matching engine.

Linear Scaling in Performance

Content Inspection Engine takes advantage of symmetric multi-threading to scale performance linearly with the number of hardware threads used. Each scan runs independently from other scans, allowing for concurrent processing of different data streams without adverse performance impact.

Multi-threading by itself is not sufficient to guarantee good scanning performance. While other software pattern matchers may indeed be multi-threaded, the contention by each thread for the shared pattern database may limit scalability.

The database in Content Inspection Engine, with its low memory footprint, coupled with the large caches in IA processors, allows each thread to scan data against a database that resides in its local cache. This dramatically reduces the amount of shared memory contention in multi-core systems, leading to a more linear progression without the traditional flattening of the performance curve as the number of threads increases.

BENCHMARKING THE INTEL XEON PROCESSOR E5-2600 SERIES

Pattern-matching performance measurements can be influenced by a number of factors. The types and numbers of signatures, the content of the incoming traffic, and the number of matches or partial matches found in the data can all affect the benchmarking results. For the results to be meaningful, the tests must use real signatures and real network traffic.

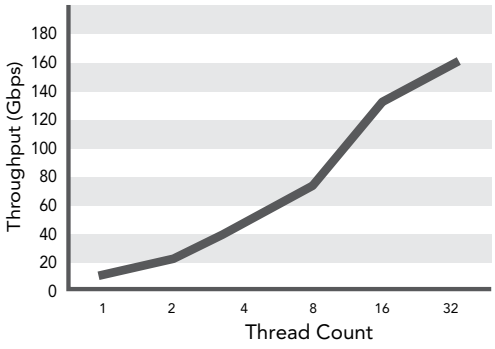
Performance benchmarking was conducted on the Content Inspection Engine library, running on a new dual-socket, quad-core (total eight cores) Intel Xeon® processor E5-2600-series-based platform using a complete set of current intrusion prevention system (IPS) signatures sourced from a leading security equipment vendor. The input was taken from real HTTP traffic captured and played back from a PCAP file. A simple application was written to read the PCAP file into memory and invoke Content Inspection Engine APIs packet by packet, simulating the behavior of a real network application such as an IPS or a web proxy. Data was matched in streaming mode for cases where the threats might span multiple packets, and in non-streaming mode for threats that would be contained within a single chunk of data.

The signature set used included multiple variants of both to-client/to-server and uniform resource identifier (URI) sets. All signatures were compiled into their run-time database in less than three seconds.

The benchmarking application specifically measured the raw pattern-matching performance, excluding the time spent in reading the PCAP file and in pre- and post-scan processing. All the data used for pattern matching was resident in-memory for this benchmark.

The results in Figure 2 show near linear scalability up to eight threads and a slight leveling off when approaching 32 threads where raw DPI scan performance tops out at 160Gbps.

Wind River Content Inspection Engine Performance on Intel Xeon Processor E5-2600 Series



Tier-1 Vendor IPS Signatures with HTTP Test Traffic	Wind River Content Inspection Engine Throughput (Gbps)					
Signature Set Type	1 Thread	2 Threads	4 Threads	8 Threads	16 Threads	18 Threads
Streaming, 69 Complex Signatures	11.9	23.3	46.3	74	132.4	159.5
Streaming, 142 Complex Signatures	6.2	12.4	24.5	43.1	81.8	95.2
Streaming, 43 Complex Signatures	3.8	7.5	14.9	25.7	48.5	56.7
Streaming, 235 Complex Signatures	1.4	2.8	5.5	10	19.1	20.8
Non-streaming, 13K Medium-Complexity Signatures	1.2	2.3	4.7	8.6	16.5	19.9
Non-streaming, 8K Medium-Complexity Signatures	2.2	4.4	8.7	16	30.6	34.4

Figure 2: Wind River Content Inspection Engine performance with current IPS signatures and real HTTP traffic

Figure 3 shows the performance progression when running the Content Inspection Engine library against a common set of signatures and data stream but varying the actual processor itself. All platforms were dual-socket, quad-core systems (eight cores). By taking the same Content Inspection Engine software library with the same APIs onto different processors in the IA family, raw scan performance can scale up and down with relative ease to match the desired equipment performance and price point.

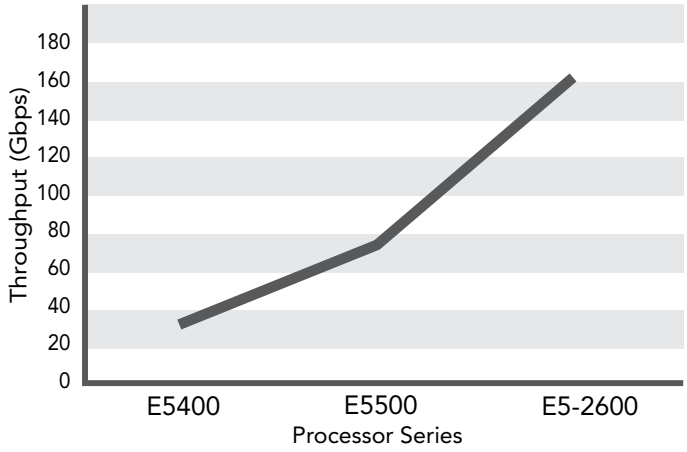


Figure 3: Scaling Wind River Content Inspection Engine performance with multi-core Intel processors

CONCLUSION

Network security equipment vendors often have to make compromises when trying to consolidate their products onto a single platform. Hardware and software designed for light-touch processing and simple packet filtering can be significantly different from that designed for advanced security applications in a large enterprise. Yet the savings in initial development costs and ongoing software maintenance make consolidation a worthwhile goal. Having parallel development organizations create similar products on different platforms is no longer viable due to market pressures to reduce costs while continuing to quickly offer new products.

Vendors are consequently looking for agile platforms that provide predictable performance and higher levels of scalability and flexibility so that the same software can be used on all products in the product family. This is only possible by adopting software-only solutions that take advantage of multi-core processing to scale up and down the product line from small sub-Gbps appliances to large multi-Gbps network equipment.

The Wind River Content Inspection Engine pattern-matching solution enables advanced security applications to scale their DPI performance linearly up to 160Gbps. The same library can be used on small appliances just as cost effectively as on large enterprise-scale network equipment merely by adjusting the number of hardware processor threads recruited.

The combination of Intel architecture processors and Wind River Content Inspection Engine enables security solution vendors to create a single platform that readily scales network throughput and satisfies the needs of different market segments without the added cost of working with multiple platforms.

intel Intelligent Systems Alliance

WIND RIVER