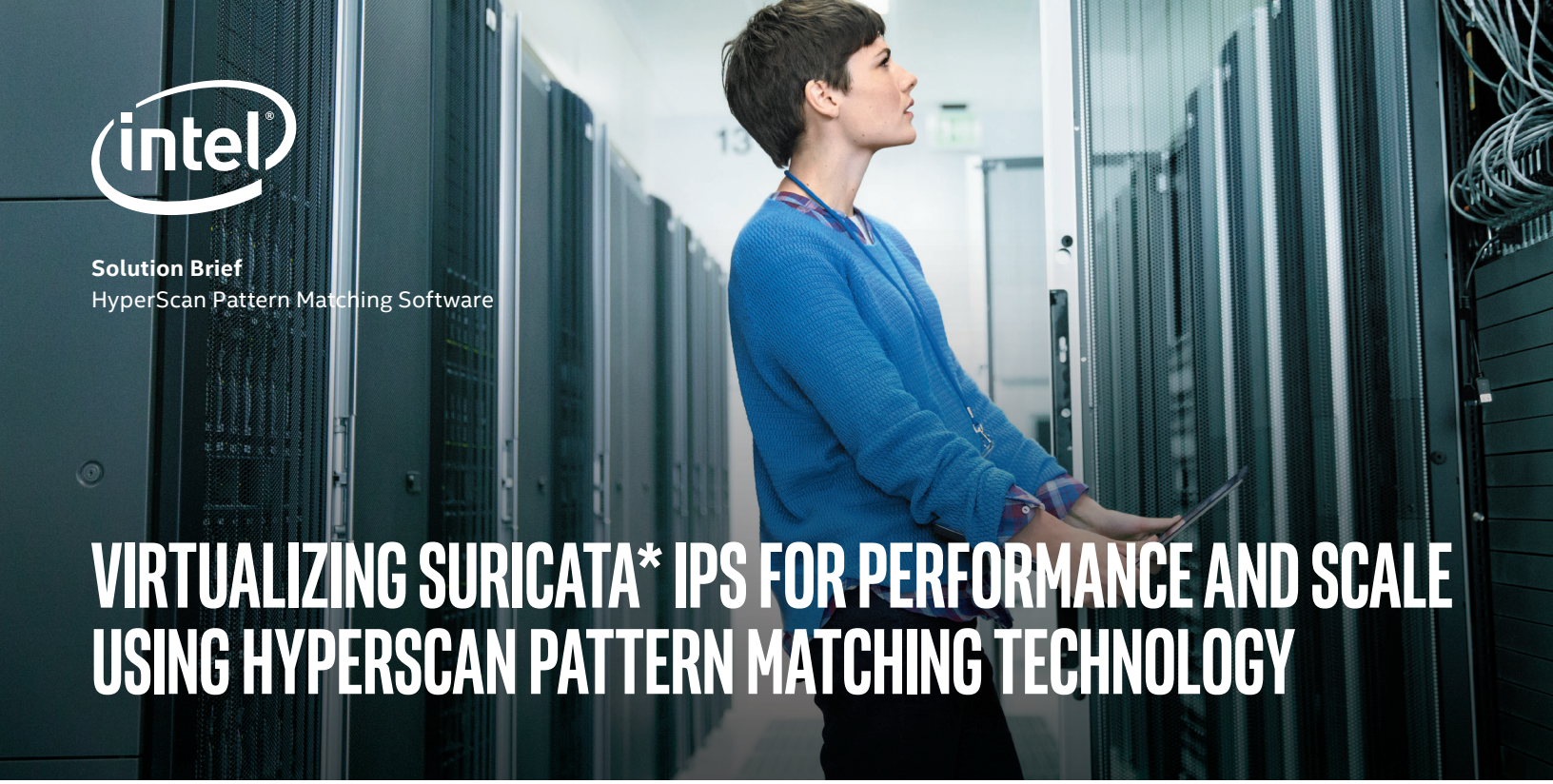

Solution Brief
HyperScan Pattern Matching Software

VIRTUALIZING SURICATA* IPS FOR PERFORMANCE AND SCALE USING HYPERSCAN PATTERN MATCHING TECHNOLOGY

Since most security applications use pattern matching, security vendors developing solutions for NFV understand the importance of delivering this function with consistent performance on virtualized computing platforms.

Executive Summary

Inundated by a large and increasing variety of proprietary hardware appliances, many network operators are encouraging the development of interoperable solutions based on high-volume, industry-standard servers. Fundamental to this transition is the use of virtualization technology to consolidate software-based network functions and services, also referred to as network functions virtualization (NFV). Not surprisingly, NFV-based solutions for telecom, enterprise, cloud environments, etc. also need to incorporate security workloads.

Since most security applications use pattern matching, security vendors developing solutions for NFV understand the importance of delivering this function with consistent performance on virtualized computing platforms. This is possible with HyperScan from Intel, which is an OS-independent, multi-threaded software pattern matching library. This paper provides performance benchmark data demonstrating the scalable pattern matching performance HyperScan delivers when combined with Suricata*, a fast-growing, open-source Intrusion Prevention and Detection (IPS/IDS) security application supported by the Open Information Security Foundation (www.oisf.org).

HyperScan Pattern Matching Library

HyperScan is a software pattern matching library that can match large groups of regular expressions against blocks or streams of data, ideal for applications that need to scan large amounts of data at high speed. HyperScan provides a simple API that is easy to integrate and is a drop-in replacement for libPCRE to deliver scan performance that is orders of magnitude better. Making it easy to implement in virtualized security solutions, HyperScan works transparently in any hypervisor environment.

When deployed on an Intel processor-based platform, HyperScan takes advantage of features such as hyper-threading, receive side scaling, and SIMD instructions to provide optimized scanning performance of over half a terabit per second on high-end Intel®

Xeon® processors. In addition, cache-rich Intel® architecture allows large matching tables to remain in cache during scanning, thus keeping memory-access overhead to a minimum.

Suricata* Overview

Suricata is an open source-based intrusion detection system (IDS), intrusion prevention system (IPS), and network security monitoring engine developed by the Open Information Security Foundation (OISF) and licensed under GPL v2. Its rule-based engine uses third-party rule sets to monitor network traffic and provide alerts to the network manager. For the solution presented here, HyperScan replaces Suricata's default multi-pattern matcher (MPM), called Aho-Corasick or "AC".

Figure 1 shows Suricata running in Host Userspace of a Linux* Host platform. Suricata spawns three packet processing threads and one management thread. On a virtualized platform, Suricata can be replicated in a series of virtual machines (VMs), which was done for the testing described in the next section.

Performance Benchmark Data

Intel and Wind River* engineers measured the throughput of an Intel Xeon processor-based platform running Suricata with HyperScan in up to ten VMs. The tests employed the Emerging Threats rule set, which included 39 rule files that have a total of 15,534 active individual rules. The tests captured network packets in a pcap file using Wireshark* while a browser simultaneously played a YouTube* video, Google*-searched "wind river", and conducted general web browsing.

The packet throughput of the Suricata detection engine was measured in five second intervals. The IPS throughput performance results shown in Figure 2 are statistically equivalent to the average throughput of all sampling periods. Labeled "Host" in the figure, the bare-metal performance of a single Suricata-HyperScan instance in a non-virtualized environment was slightly over 9,000 Mbps. VM1 through VM10 correspond to one to ten instances running simultaneously, with throughput ranging from approximately 5,700 to 8,200 Mbps.

The performance data raises three important points:

- Figure 2 shows near-linear VM performance baselined (green line) at 6 Gbps.
- Compared to running Suricata on bare-metal, instantiating one to ten VMs resulted in a consistent.
- Adding a VM did not necessarily reduce the throughput of the other VMs.

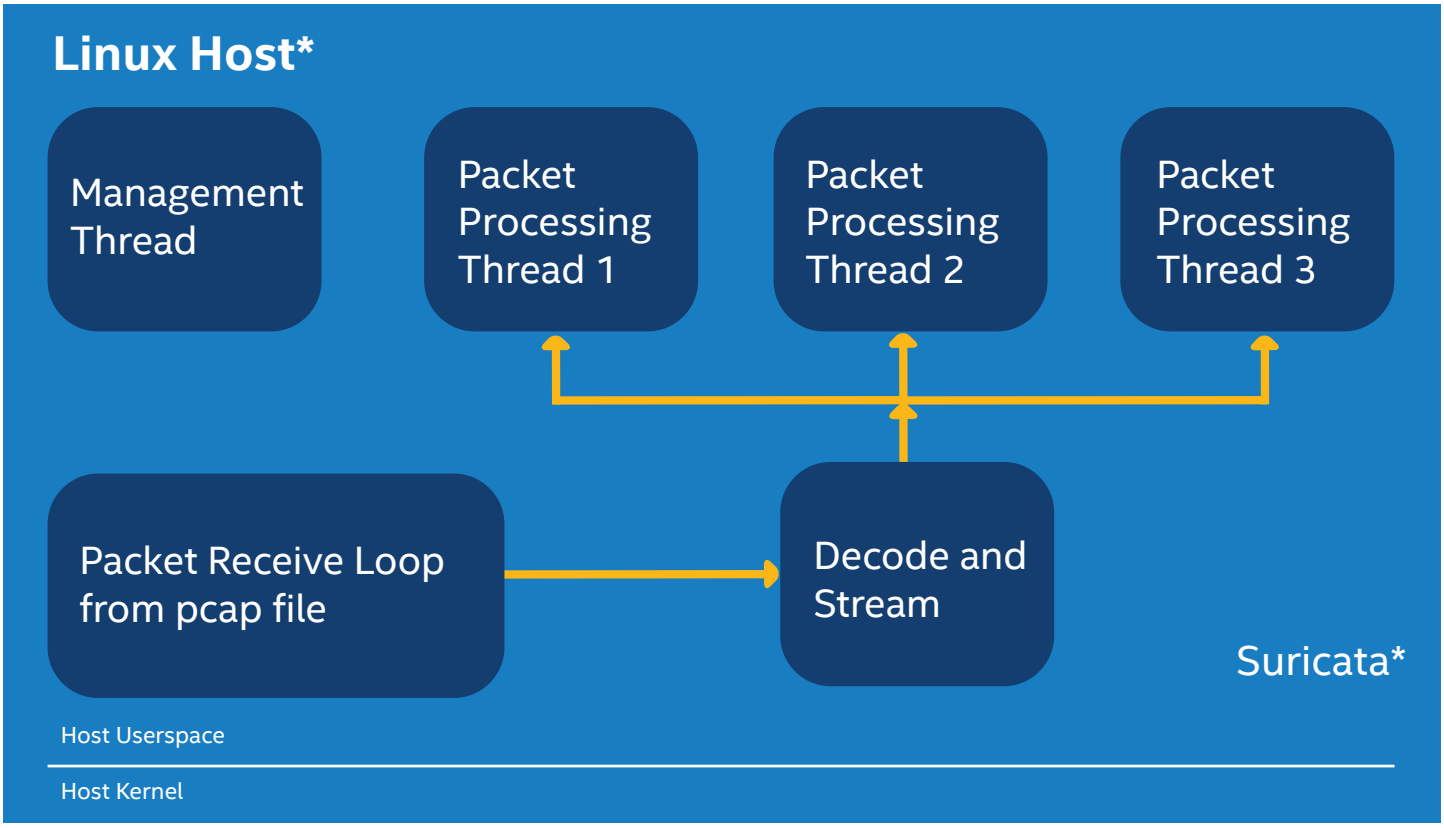

Figure 1. Suricata* Block Diagram

The performance variations between the VM instances mainly reflect the fact that the VM instances are user-space software applications scheduled by the host Linux kernel. In the benchmarking test, core affinity is not used by the Suricata configuration.

The number of VM instances was limited to ten due the following factors:

1. The host system has 64 GB of RAM
2. Each VM instance uses 5 GB of RAM
3. The host system has a total of 72 hyper-threading cores
4. Each VM instance uses six hyper-threading cores

Therefore, the maximum number of VM instances running simultaneously is impacted by the amount of system memory and the core count.

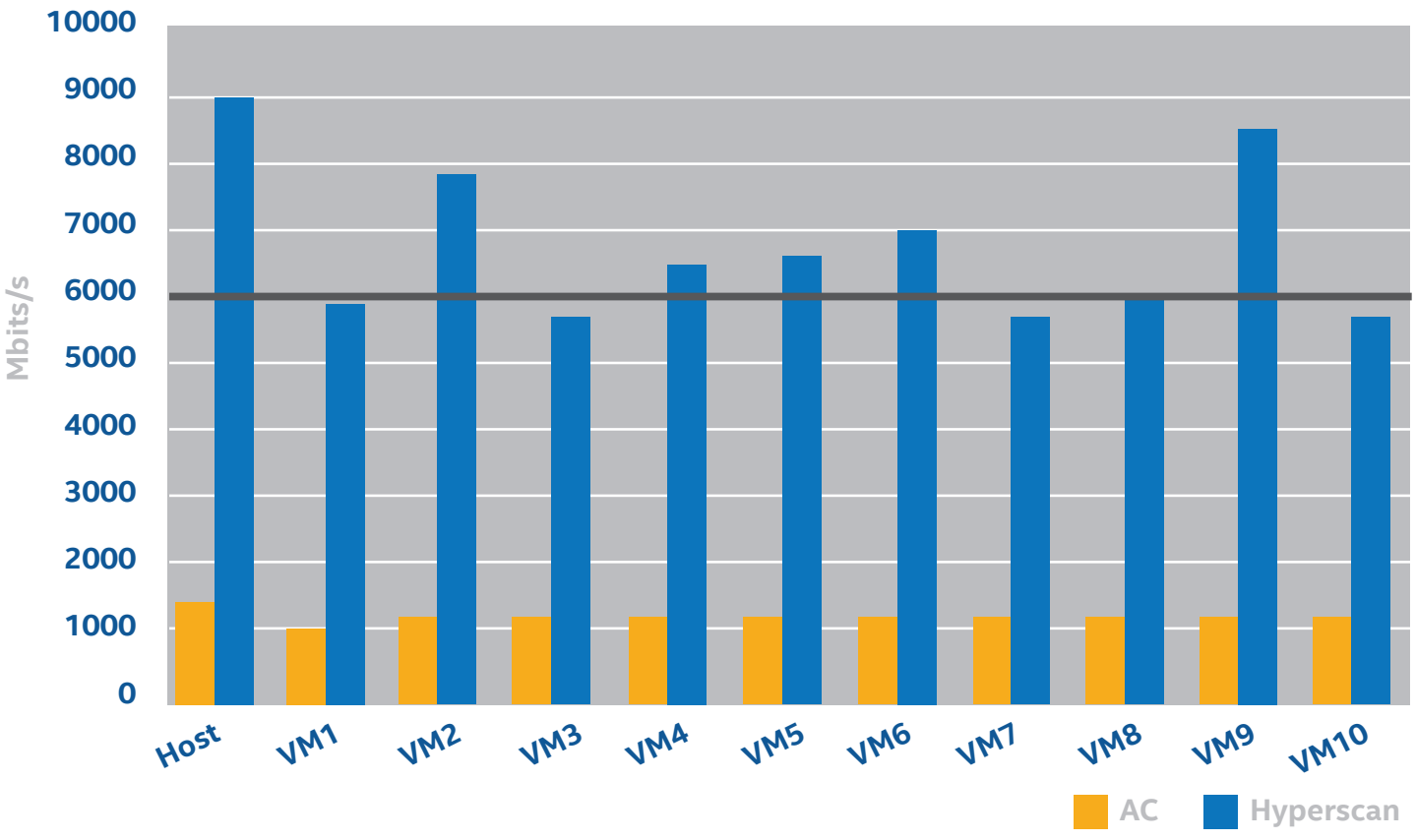

Figure 2. Comparison of AC and HyperScan Performance on a Virtualized Platform

TEST SETUP

Table 1 contains more details about the bench setup.

TEST SETUP	
Hardware	Platform: Intel® Server Board S2600WT Family • Dual Intel® Xeon® processor E5-2699 v3 • 18 cores/each (36 cores total) • 64 GB System Memory • Intel® Hyper-Threading Technology enabled (72 threads total)
Software	Linux* Guest KVM Configuration • Copy host CPU configuration • Three physical cores • 5 GB DRAM • 4 GB vDisk • CentOS 7 for x86_64 • Up to 10 VMs running on the host

Table 1. Test Setup Details

Virtualization-Ready

With the advanced capability of the Intel architecture coupled with HyperScan high pattern matching performance, network security vendors looking to deliver NFV-based security solutions can now deliver virtualized security solutions that can extensively scale in any operating environment while delivering high throughput performance that is predictable.

This solution is ideal for NFV/SDN-based equipment, offering a highly flexible and scalable content inspection solution that runs transparently in any hypervisor environment. HyperScan performance and functionality, whether virtualized or non-virtualized, scales linearly on a per core/thread basis on Intel® processors.

[Δ] Intel® processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: http://www.intel.com/products/processor_number.

[1] Source: http://www.anandtech.com/show/6993/intel-iris-pro-5200-graphics-review-core-i74950hq-tested/18.

[2] Intel® technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

[3] Source: Intel testing.

[4] Intel® technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

[5] Drivers available at: downloadcenter.intel.com

[6] No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.