White Paper

Video Analytics
AI Vision/Deep Learning

Accelerate Video Analytics Performance

ST Engineering leverages Intel® technology to deliver breakthrough performance efficiency for AI-powered video analytics

Intel team authors

Nee Yuan Kuok
IOTG Platform Application Engineer

Gary Koay
IOTG Platform Application Engineer

Brandon Koh
Industry Technical Specialist

ST Engineering team authors

Hui Ming Liew
Product Manager (Video Analytics)

Chinthaka Gamanayake
Machine Learning Senior Software Engineer (Video Analytics)

Yixin Chang
Machine Learning Software Engineer (Video Analytics)

Table of contents

Executive summary

ST Engineering is a global technology, defense, and engineering group with a diverse portfolio of businesses across the aerospace, smart city, defense, and public security segments. Its smart city technologies help cities prepare for a more connected, resilient, and sustainable urban future by addressing connectivity, mobility, security, infrastructure, and environmental needs.

ST Engineering's AGIL Video Analytics Platform (AGIL VAP) is an open architecture solution based on cloud technology that supports a range of video analytics engines for smart city applications involving recognition of people, vehicles, and objects. The scale and complexity of such systems may range from 100 to 10,000 video streams, with deployment of computing resources either in the cloud or at the edge of the network, depending on customer requirements. This flexible, single-platform solution enables customers to scale up or add video analytics engines to meet changing operational requirements.

To achieve industry-leading performance efficiency for AI-powered video analytics, ST Engineering leveraged deep learning inference models included with the Intel® Distribution of OpenVINO™ toolkit, as well as artificial intelligence (AI) acceleration capabilities built into 3rd Generation Intel® Xeon® Scalable processors.

Benchmarking data shows an over 12x increase in performance efficiency with a 90 percent decrease in power consumption when using the OpenVINO toolkit on 3rd Generation Intel Xeon Scalable processors, compared to TensorFlow on 2nd Generation Intel® Xeon® Scalable processorss.[1]

These results demonstrate the potential for developers of smart city video analytics (VA) solutions to deliver highly cost-effective AI performance by leveraging Intel® technology—while delivering the architectural flexibility needed to balance lower capital expenditures (CapEx) with reasonable long-term operating expenses (OpEx) over the life of the system.

Video analytics in smart cities

As city governments, transportation authorities, and public safety agencies adopt smart city solutions and technologies, they unlock new tools, practices, and insights that can improve responsiveness, services, and quality of life for their citizens.

To help make urban life safer, more efficient, and more sustainable, smart cities use data from embedded, intelligent technologies, including cameras and Internet of Things (IoT) sensors with integrated compute to collect and analyze data from locations around the city in near-real time. Decision-makers then use the knowledge obtained from this data to monitor and protect the city's properties, capital, and services.

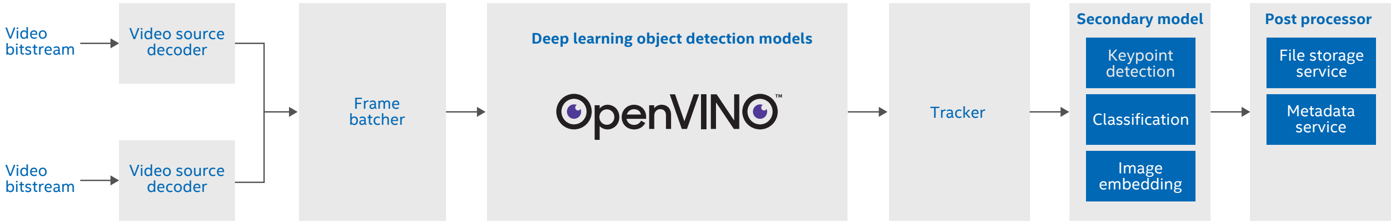


Figure 1. System architecture of ST Engineering's video analytics engine.

Increasing threats to global and domestic security, along with costs associated with hiring security personnel, are driving demand for VA, which uses AI deep learning inference to recognize people, vehicles, and objects in video footage. The use of VA reduces the hefty cost and tedious process of using human operators to view large amounts of video data for abnormal incidents and behaviors.

The use of this technology must always be paired with an emphasis on data privacy and security, as well as consent and civil rights, by the relevant government, town council, business, or operator. Regardless of local laws, Intel takes a firm stance on eliminating bias in the collection and analysis of data. Use of data must be restricted to public safety and security purposes under a system of checks and balances. In many cases, subjects should be made aware they are being recorded and given a chance to opt in or out.

The availability of cost-effective high performance computing power, along with software toolkits optimized for deep learning inference, makes performing VA computation at the network edge increasingly feasible. VA at the edge lowers bandwidth demands and decreases communication delays, enabling end users to make quicker decisions in sensitive circumstances. Although data travels from endpoints to the cloud in 150 to 200 milliseconds, it takes only 10 milliseconds from endpoints to the edge, enabling more-efficient detection and reaction.[2]

Additionally, public and private sector players responsible for developing smart city infrastructure face pressure to keep CapEx as low as possible. From this perspective, centralized deployment of cloud-based computing resources may be advantageous compared to the CapEx of deploying numerous AI-capable IoT devices around the city. On the other hand, cloud-based VA can be challenging to scale, with ongoing OpEx costs contributing to a higher total cost of ownership (TCO) over time.

From a privacy and security perspective, the implementation of a video analytics system based on a multitier architecture enables the segregation of different data layers to reduce the threat of intrusion and loss. When using public cloud infrastructure, certain use cases (e.g., vehicle and traffic related) may be deemed less sensitive, meriting a less-stringent security architecture.

The ideal VA solution for smart city applications, then, is one that lowers the cost of deployment while giving customers the flexibility to leverage any computing resources available—in the cloud, at the edge, or both—and to adjust that balance to meet a range of scales, budgets, privacy and security needs, and business requirements.

Potential customers

Potential customers for ST Engineering's AGIL VAP include public safety and security agencies, security companies, building owners, mall operators, city planners, and municipal governments.

System architecture

ST Engineering's AGIL VAP is an open architecture cloud-based system that manages multiple video analytics engines on a single platform. AGIL VAP is a full-stack solution that enables operators to seamlessly execute video analytics jobs, generating the necessary alerts to shorten the detection and response cycle.

The solution enables customers to scale up or add VA engines to meet changing operational requirements. VA engines can share a common pool of computing resources, enabling use of the right engine to achieve optimal insights for the application at hand. VA engines can also provide capabilities from multiple types of hardware, enabling them to use all types of computing resources. AGIL VAP is customizable and can be agnostic to different hardware architectures, avoiding compatibility problems across engines, which is a common issue.

As shown in Figure 1, AGIL VAP is designed based on a modern software architecture of microservices coupled with a library of optimized models that go through an automated machine learning operations (MLOps) pipeline and workflow. The codebase for the AI/ML models and the video processing pipeline are optimized at the silicon level for high performance and reliability.

AGIL VAP takes in video streams through the real-time streaming protocol (RTSP) using video decoders built into the Intel® CPU. The data then undergoes a deep learning–based object detection algorithm based on models included with the Intel Distribution of OpenVINO toolkit, which optimizes deep learning through Vector Neural Network Instructions (VNNI) and Intel® Advanced Vector Extensions 512 (Intel® AVX-512) operations. The resulting model is then passed to other microservices in AGIL VAP.

Software development and optimization

ST Engineering originally used TensorFlow to build a baseline generic pipeline that worked across numerous silicon architectures, including Intel® x86. The engineers decided to use the Intel Distribution of OpenVINO toolkit to explore cost optimization without the need for an additional graphics accelerator to reduce potential points of failure during deployment.

By using Intel® performance optimization tools to analyze and tune reference models provided with the OpenVINO toolkit, ST Engineering achieved substantial improvements in performance efficiency and power consumption. In addition, they were able to develop a solution more quickly than would otherwise have been possible.

Intel Distribution of OpenVINO toolkit

OpenVINO is a comprehensive toolkit for quickly developing applications and solutions that solve various AI-powered tasks, such as VA. Based on the latest generations of artificial neural networks, including convolutional neural networks (CNNs) and recurrent and attention-based networks, the toolkit maximizes performance by extending computer vision and nonvision workloads across Intel® hardware.

ST Engineering's OpenVINO-optimized pipeline enables the use of a range of accelerators and Intel CPU architectures with minimal code changes. OpenVINO enables developers to deploy the same application across combinations of host processors, accelerators, and environments, including CPUs, GPUs, VPUs, FPGAs, whether on-premises or on device, in the browser or in the cloud.

OpenVINO's Open Model Zoo provided ST Engineering with quantized pretrained models for VA, offering targeted cost and performance and reducing potential points of failure. Data type conversion from fp32 to int8 was also provided by a model download from Open Model Zoo. In addition, OpenVINO provides a transfer-learning toolkit to accommodate specific retraining capabilities.

OpenVINO is a well-documented framework with extensive tools and examples, enabling ST Engineering's team to get up to speed quickly and dive deep into optimization. Intel provided access to experts in the VA domain, including the ability to speak directly to teams that developed the Intel hardware and software they were using. ST Engineering worked with Intel's Internet of Things Group (IOTG) in Singapore and Malaysia, and with OpenVINO developers in India and China.

CPU upgrade

ST Engineering set out to benchmark AGIL VAP on 3rd Gen Intel Xeon Scalable processors in order to take advantage of built-in acceleration for training and inference workloads as well as increased cache architecture and memory bandwidth and channels. Upgrading from 2nd Gen to 3rd Gen Intel Xeon Scalable processors brought significant performance improvements.[1]

Intel® Deep Learning Boost (Intel® DL Boost) with VNNI

The second generation of Intel Xeon Scalable processors introduced a collection of features designed to accelerate AI/DL inference, packaged together as Intel Deep Learning Boost. These features include VNNI, which increases throughput for inference applications with support for int8 convolutions by combining multiple machine instructions from previous generations into one machine instruction.

Based on Intel AVX-512, Intel DL Boost VNNI delivers a significant performance improvement by combining three instructions into one—maximizing the use of compute resources, better utilizing the cache, and avoiding potential bandwidth bottlenecks.

ST Engineering's results leveraged special instructions, such as Intel AVX-512 and VNNI, which resulted in an uplift in instructions per cycle (IPC), with the process nodes providing better high-frequency sustaining when Intel AVX-512 and VNNI were used.

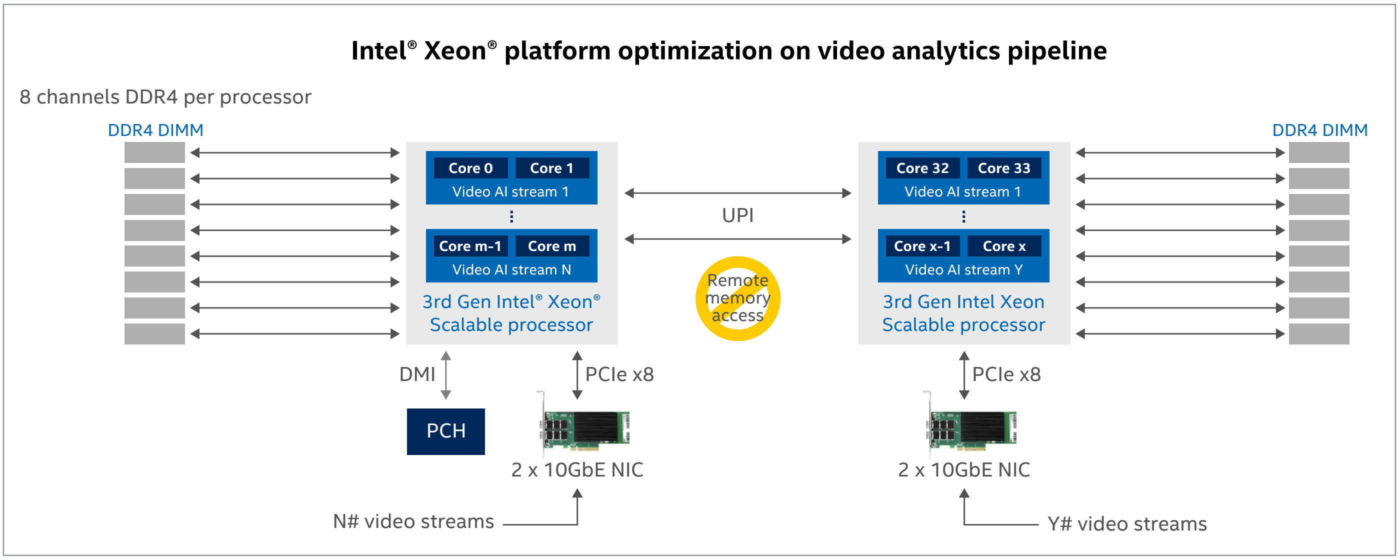
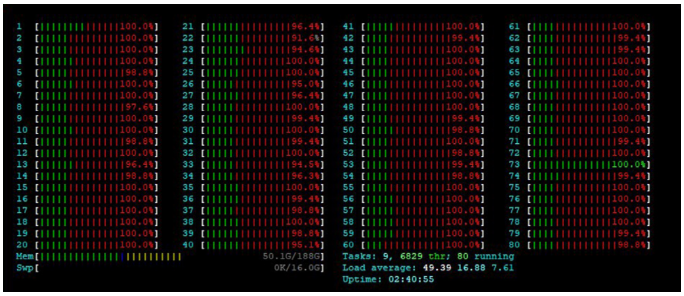
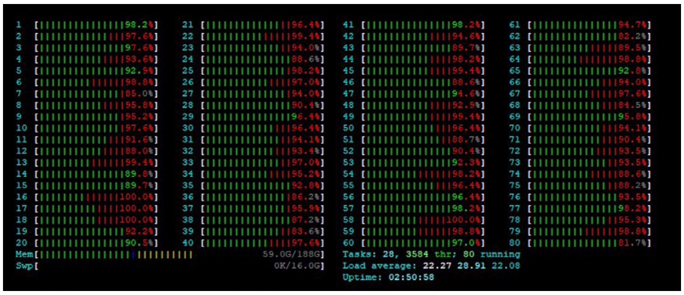


Figure 2. Video analytics pipeline of ST Engineering's video analytics engine.



```
1  [||||||||||||||100.0%]   21 [|||||||||||||||96.4%]   41 [||||||||||||||100.0%]   61 [||||||||||||||100.0%]
2  [||||||||||||||100.0%]   22 [|||||||||||||||91.6%]   42 [|||||||||||||||99.4%]   62 [|||||||||||||||99.4%]
3  [||||||||||||||100.0%]   23 [|||||||||||||||94.6%]   43 [||||||||||||||100.0%]   63 [|||||||||||||||99.4%]
4  [||||||||||||||100.0%]   24 [||||||||||||||100.0%]   44 [||||||||||||||100.0%]   64 [||||||||||||||100.0%]
5  [|||||||||||||||98.8%]   25 [||||||||||||||100.0%]   45 [||||||||||||||100.0%]   65 [||||||||||||||100.0%]
6  [||||||||||||||100.0%]   26 [|||||||||||||||95.0%]   46 [||||||||||||||100.0%]   66 [||||||||||||||100.0%]
7  [||||||||||||||100.0%]   27 [|||||||||||||||96.4%]   47 [||||||||||||||100.0%]   67 [||||||||||||||100.0%]
8  [|||||||||||||||97.6%]   28 [||||||||||||||100.0%]   48 [||||||||||||||100.0%]   68 [||||||||||||||100.0%]
9  [||||||||||||||100.0%]   29 [|||||||||||||||99.4%]   49 [|||||||||||||||99.4%]   69 [||||||||||||||100.0%]
10 [||||||||||||||100.0%]   30 [||||||||||||||100.0%]   50 [|||||||||||||||98.8%]   70 [||||||||||||||100.0%]
11 [|||||||||||||||98.8%]   31 [|||||||||||||||99.4%]   51 [||||||||||||||100.0%]   71 [|||||||||||||||99.4%]
12 [||||||||||||||100.0%]   32 [||||||||||||||100.0%]   52 [|||||||||||||||98.8%]   72 [||||||||||||||100.0%]
13 [|||||||||||||||96.4%]   33 [|||||||||||||||94.5%]   53 [|||||||||||||||99.4%]   73 [||||||||||||||100.0%]
14 [|||||||||||||||98.8%]   34 [|||||||||||||||96.3%]   54 [|||||||||||||||98.8%]   74 [|||||||||||||||99.4%]
15 [||||||||||||||100.0%]   35 [||||||||||||||100.0%]   55 [||||||||||||||100.0%]   75 [||||||||||||||100.0%]
16 [||||||||||||||100.0%]   36 [|||||||||||||||99.4%]   56 [||||||||||||||100.0%]   76 [||||||||||||||100.0%]
17 [||||||||||||||100.0%]   37 [|||||||||||||||98.8%]   57 [||||||||||||||100.0%]   77 [||||||||||||||100.0%]
18 [||||||||||||||100.0%]   38 [||||||||||||||100.0%]   58 [||||||||||||||100.0%]   78 [||||||||||||||100.0%]
19 [||||||||||||||100.0%]   39 [|||||||||||||||98.8%]   59 [||||||||||||||100.0%]   79 [|||||||||||||||99.4%]
20 [||||||||||||||100.0%]   40 [|||||||||||||||95.1%]   60 [||||||||||||||100.0%]   80 [|||||||||||||||98.8%]
Mem[|||||||||||||||||||||||||             50.1G/188G]   Tasks: 9, 6829 thr; 80 running
Swp[                                        0K/16.0G]   Load average: 49.39 16.88 7.61
                                                        Uptime: 02:40:55
```

Figure 3. Preoptimization pipeline process scheduling.



```
1  [|||||||||||||||98.2%]   21 [|||||||||||||||96.4%]   41 [|||||||||||||||98.2%]   61 [|||||||||||||||94.7%]
2  [|||||||||||||||97.6%]   22 [|||||||||||||||99.4%]   42 [|||||||||||||||94.6%]   62 [|||||||||||||||82.2%]
3  [|||||||||||||||97.6%]   23 [|||||||||||||||94.0%]   43 [|||||||||||||||89.7%]   63 [|||||||||||||||89.5%]
4  [|||||||||||||||93.6%]   24 [|||||||||||||||88.6%]   44 [|||||||||||||||98.2%]   64 [|||||||||||||||98.8%]
5  [|||||||||||||||92.9%]   25 [|||||||||||||||98.2%]   45 [|||||||||||||||99.4%]   65 [|||||||||||||||92.8%]
6  [|||||||||||||||98.8%]   26 [|||||||||||||||97.0%]   46 [|||||||||||||||88.6%]   66 [|||||||||||||||94.0%]
7  [|||||||||||||||85.0%]   27 [|||||||||||||||94.0%]   47 [|||||||||||||||94.6%]   67 [|||||||||||||||97.6%]
8  [|||||||||||||||95.8%]   28 [|||||||||||||||90.4%]   48 [|||||||||||||||92.9%]   68 [|||||||||||||||84.5%]
9  [|||||||||||||||95.2%]   29 [|||||||||||||||96.4%]   49 [|||||||||||||||99.4%]   69 [|||||||||||||||95.8%]
10 [|||||||||||||||97.6%]   30 [|||||||||||||||96.4%]   50 [|||||||||||||||96.4%]   70 [|||||||||||||||94.1%]
11 [|||||||||||||||91.6%]   31 [|||||||||||||||94.1%]   51 [|||||||||||||||88.7%]   71 [|||||||||||||||90.4%]
12 [|||||||||||||||88.0%]   32 [|||||||||||||||93.4%]   52 [|||||||||||||||90.4%]   72 [|||||||||||||||93.5%]
13 [|||||||||||||||99.4%]   33 [|||||||||||||||97.0%]   53 [|||||||||||||||92.3%]   73 [|||||||||||||||93.5%]
14 [|||||||||||||||89.8%]   34 [|||||||||||||||95.2%]   54 [|||||||||||||||98.2%]   74 [|||||||||||||||88.6%]
15 [|||||||||||||||89.7%]   35 [|||||||||||||||92.8%]   55 [|||||||||||||||96.4%]   75 [|||||||||||||||88.2%]
16 [||||||||||||||100.0%]   36 [|||||||||||||||86.2%]   56 [|||||||||||||||96.4%]   76 [|||||||||||||||93.5%]
17 [||||||||||||||100.0%]   37 [|||||||||||||||95.9%]   57 [|||||||||||||||98.2%]   77 [|||||||||||||||98.2%]
18 [||||||||||||||100.0%]   38 [|||||||||||||||87.2%]   58 [||||||||||||||100.0%]   78 [|||||||||||||||95.3%]
19 [|||||||||||||||92.2%]   39 [|||||||||||||||83.6%]   59 [|||||||||||||||98.8%]   79 [|||||||||||||||98.8%]
20 [|||||||||||||||90.5%]   40 [|||||||||||||||97.6%]   60 [|||||||||||||||97.0%]   80 [|||||||||||||||81.7%]
Mem[||||||||||||||||||||||||||||           59.0G/188G]   Tasks: 28, 3584 thr; 80 running
Swp[                                        0K/16.0G]   Load average: 22.27 28.91 22.08
                                                        Uptime: 02:50:58
```

Figure 4. Postoptimization pipeline process scheduling.

The video analytics pipeline is an asynchronous process that leverages the multicore topology and noninclusive last-level cache (LLC) features of the 3rd Generation Intel Xeon Scalable processor.

The asynchronous process generally consumes a certain amount of CPU cycle on the OS scheduler for context switching. With unoptimized video pipeline process scheduling, the overhead of kernel context switching overwhelmed the CPU, as shown in Figure 3, resulting in lower efficiency and less data being processed by the application. (The report output shown in Figures 3 and 4 is from htop, an interactive process viewer.)

Pinning the core and allocating local memory for the application prevents the kernel from performing excessive context switching. Remote socket memory access, which tends to introduce CPU stall while waiting for memory transfer via Intel® Ultra Path Interconnect (Intel® UPI), is reduced, as shown in Figure 4.

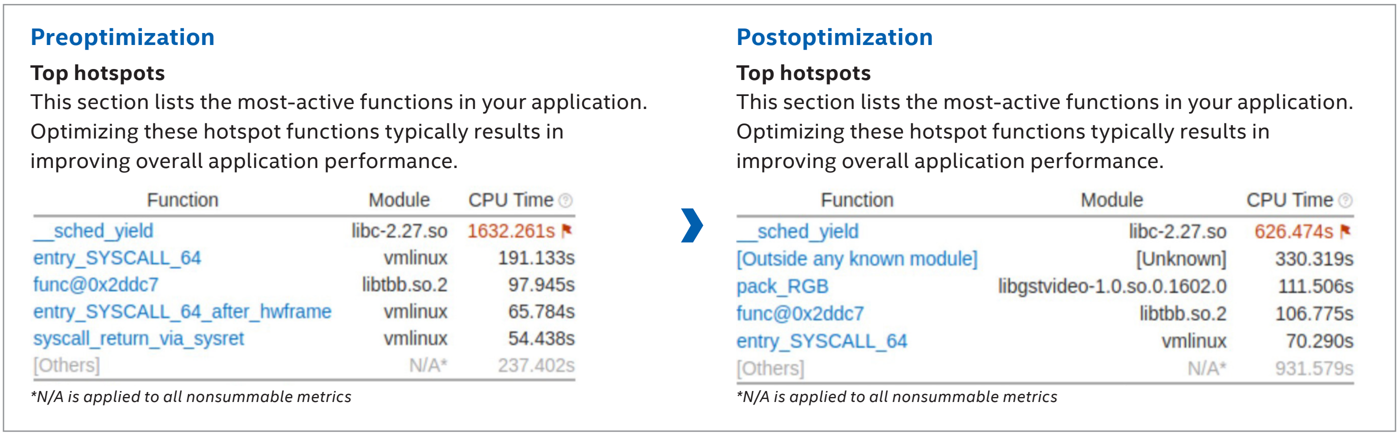
The user application is allocated to manage and process video streams local to the network interface controller (NIC) card to prevent excessive cross-socket memory transfer, which might reduce the efficiency of the caching and home agent (CHA).

Intel® VTune™ Profiler

The ST Engineering team used Intel VTune Profiler to optimize application performance, system performance, and system configuration for AGIL VAP. Using Intel VTune, core utilization headroom was catered for additional demand requests.

Core pinning reduces core switching by restricting computations within certain virtual cores. This has a large performance effect on the GStreamer pipeline within the VA engine, which, as a multithreaded framework, otherwise automatically distributes thread-based workloads to cores across the entire server.

A workload of 40 video analytics pipelines, each processing a video, was measured to produce the results shown in Figure 5. CPU workloads on pinned cores require much less computation to do the same amount of work because they use cores more effectively, whereas unpinned cores need to wait for the thread pool manager to instantiate context switching before they can run their workloads.



Preoptimization

Top hotspots

This section lists the most-active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time
__sched_yield	libc-2.27.so	1632.261s
entry_SYSCALL_64	vmlinux	191.133s
func@0x2ddc7	libtbb.so.2	97.945s
entry_SYSCALL_64_after_hwframe	vmlinux	65.784s
syscall_return_via_sysret	vmlinux	54.438s
[Others]	N/A*	237.402s

*N/A is applied to all nonsummable metrics

Postoptimization

Top hotspots

This section lists the most-active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

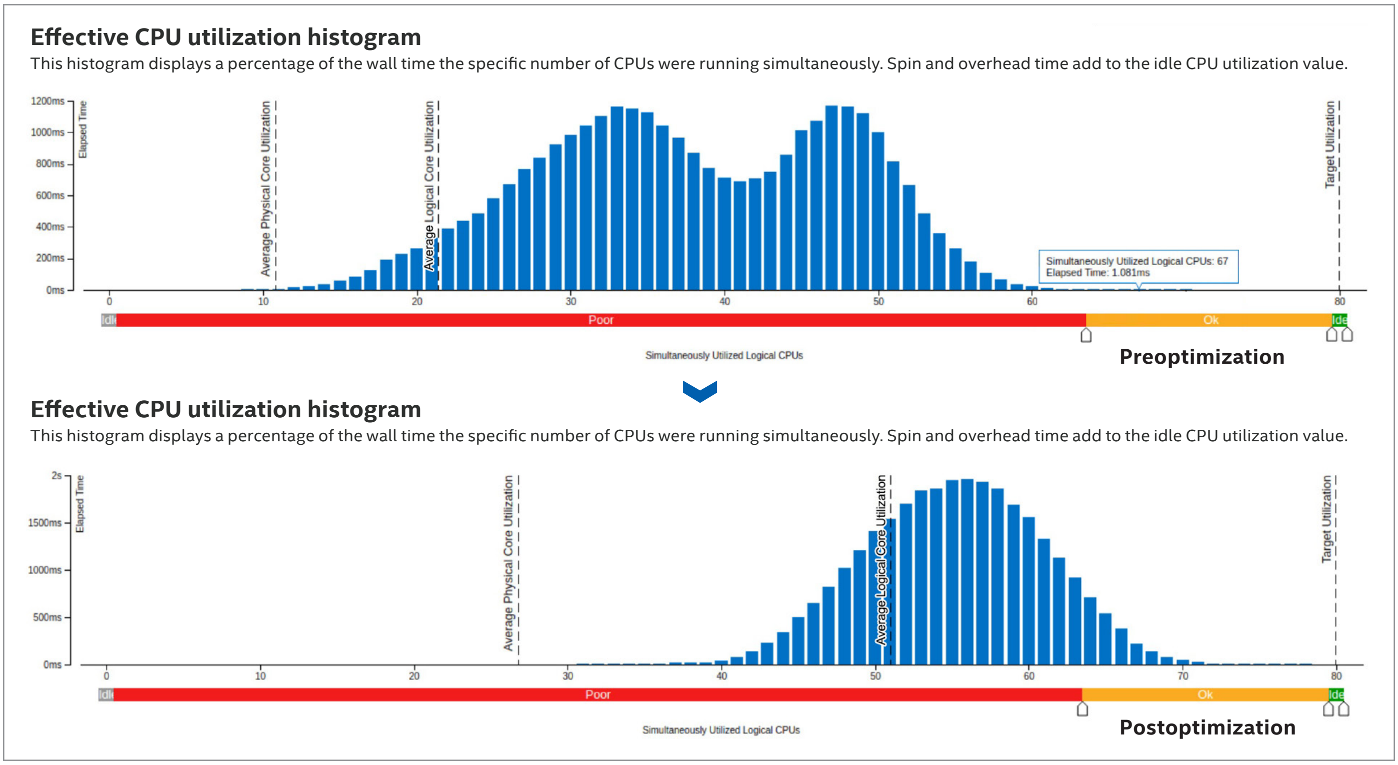
Function	Module	CPU Time
__sched_yield	libc-2.27.so	626.474s
[Outside any known module]	[Unknown]	330.319s
pack_RGB	libgstvideo-1.0.so.0.1602.0	111.506s
func@0x2ddc7	libtbb.so.2	106.775s
entry_SYSCALL_64	vmlinux	70.290s
[Others]	N/A*	931.579s

*N/A is applied to all nonsummable metrics

Figure 5. Summary of Intel® VTune™ Profiler analysis.

Figure 5 shows most of the active functions in the application. In preoptimization analysis, the CPU capacity consumed by kernel scheduler is about 68 percent of the total, resulting in lower user-space application performance. This is due to the process execution span across dual sockets, showing overwhelmed UPI with memory transfer and core stalls, which incur wait time on the threads.

In postoptimization analysis, the scheduler CPU use is reduced by 160 percent, though it is still the most active function due to the asynchronous nature of the video analytic pipeline.[1]



Effective CPU utilization histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and overhead time add to the idle CPU utilization value.

Effective CPU utilization histogram

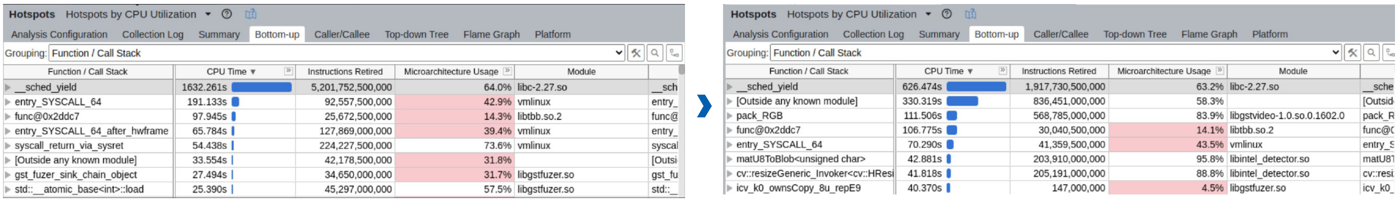
This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and overhead time add to the idle CPU utilization value.

2s
1500ms
1000ms
500ms
0ms
Elapsed Time
Average Physical Core Utilization
Average Logical Core Utilization
Target Utilization
0
10
20
30
40
50
60
70
80
Idle
Poor
Ok
Ide
Simultaneously Utilized Logical CPUs
Postoptimization

Figure 6. Histogram of Intel® VTune™ Profiler analysis.

Figure 6 is a histogram showing the number of simultaneously used logical CPU cores. In the preoptimization histogram, the platform uses only 30 to 50 cores out of 80 simultaneously within the 30 seconds of platform performance sampling, with the average number of cores being used at the same time around 21.

In the postoptimization histogram, the average logical core utilization is improved to about 51—more than double preoptimization. This demonstrates that the platform optimization technique can be effectively applied to the multiprocess video pipeline.



Grouping: Function / Call Stack

Function / Call Stack	CPU Time ▼	Instructions Retired	Microarchitecture Usage	Module	
__sched_yield	1632.261s	5,201,752,500,000	64.0%	libc-2.27.so	__sch
entry_SYSCALL_64	191.133s	92,557,500,000	42.9%	vmlinux	entry_
func@0x2ddc7	97.945s	25,672,500,000	14.3%	libtbb.so.2	func@
entry_SYSCALL_64_after_hwframe	65.784s	127,869,000,000	39.4%	vmlinux	entry_
syscall_return_via_sysret	54.438s	224,227,500,000	73.6%	vmlinux	syscal
[Outside any known module]	33.554s	42,178,500,000	31.8%		[Outsi
gst_fuzer_sink_chain_object	27.494s	34,650,000,000	31.7%	libgstfuzer.so	gst_fu
std::__atomic_base<int>::load	25.390s	45,297,000,000	57.5%	libgstfuzer.so	std::_

Grouping: Function / Call Stack

Function / Call Stack	CPU Time ▼	Instructions Retired	Microarchitecture Usage	Module	
__sched_yield	626.474s	1,917,730,500,000	63.2%	libc-2.27.so	__sche
[Outside any known module]	330.319s	836,451,000,000	58.3%		[Outsid
pack_RGB	111.506s	568,785,000,000	83.9%	libgstvideo-1.0.so.0.1602.0	pack_R
func@0x2ddc7	106.775s	30,040,500,000	14.1%	libtbb.so.2	func@(
entry_SYSCALL_64	70.290s	41,359,500,000	43.5%	vmlinux	entry_S
matU8ToBlob<unsigned char>	42.881s	203,910,000,000	95.8%	libintel_detector.so	matU8T
cv::resizeGeneric_Invoker<cv::HResi	41.818s	205,191,000,000	88.8%	libintel_detector.so	cv::resi
icv_k0_ownsCopy_8u_repE9	40.370s	147,000,000	4.5%	libgstfuzer.so	icv_k0_

Figure 7. Bottom-up function call analysis from Intel® VTune™ Profiler.

Figure 7 shows the VTune function called bottom-up analysis. Preoptimization analysis shows the kernel scheduling using more than 68 percent of CPU time, showing room for improvement using core pinning to reduce remote socket memory access and UPI memory transfer.[1]

In the postprocessing analysis, the top active processes are occupied by the actual user space workload, such as the Intel® Threading Building Blocks (Intel® TBB) library and GStreamer, instead of being crowded out by kernel process overhead.

Figure 8 summarizes the results of the optimization performed with Intel VTune, showing improvements in core utilization efficiency and kernel overhead percentage.[1]

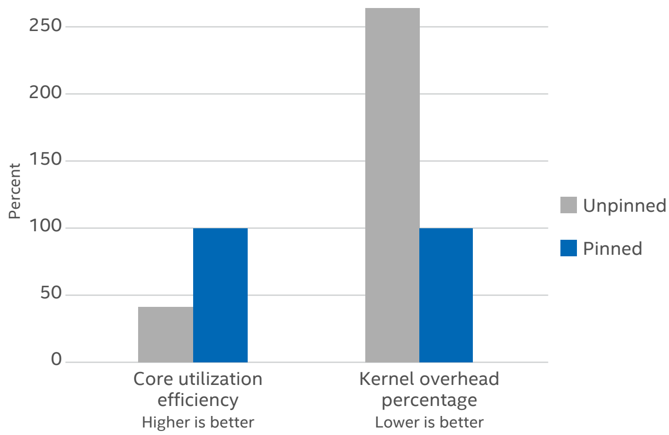


Figure 8. Summary of Intel® VTune™ optimization.

Results

ST Engineering collaborated with Intel to measure performance benchmarks on the AGIL VAP code. By leveraging deep learning inference models included with the Intel Distribution of OpenVINO toolkit, along with AI acceleration capabilities built into 3rd Generation Intel Xeon Scalable processors, ST Engineering was able to achieve industry-leading performance efficiency for AI-powered video analytics.

As shown in Figure 9, benchmarking data showed a 5x improvement in performance efficiency by moving from TensorFlow to OpenVINO on 2nd Generation Xeon Scalable processors with fp32 and a 2x improvement in moving from 2nd Generation to 3rd Generation Intel Xeon Scalable processors with int8, for an overall improvement of over 12x.[1]

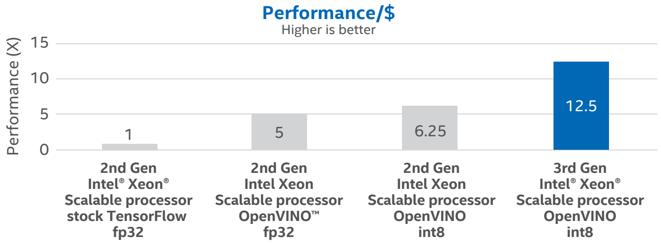


Figure 9. Performance efficiency benchmarking results for ST Engineering's AGIL Video Analytics Platform ($/FPS).[3]

Performance efficiency results are expressed in dollars per frame per second ($/FPS). $/FPS calculations are based on standardized hardware costs from a server vendor (including CPU, RAM, basic storage, networking, and chassis), divided by the number of frames that can be processed through the AGIL VAP software per second. This gives a good estimate of the cost of ownership for a large enough deployment to saturate the processing power of the server CPUs.

Figure 10 shows an 80 percent drop in moving from TensorFlow to OpenVINO on 2nd Gen Intel Xeon Scalable processors with fp32 and a nearly 50 percent drop in moving from 2nd Gen to 3rd Gen Intel Xeon Scalable processors with int8, for an overall drop in power consumption of 90 percent.[1] Power consumption is expressed in watts per frames per second, or W/FPS.

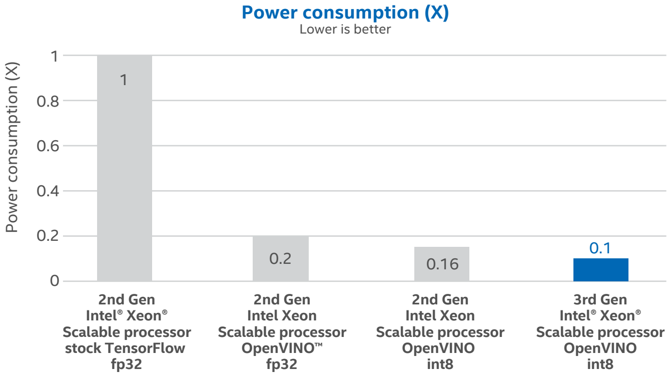


Figure 10. Power consumption benchmarking results for ST Engineering's AGIL Video Analytics Platform (W/FPS).

Conclusion

In this paper, we have detailed the methods used by ST Engineering to achieve industry-leading improvements in performance efficiency for their Video Analytics Platform.

Benchmarking data shows an over 12x increase in performance efficiency with a 90 percent decrease in power consumption when using OpenVINO toolkit on 3rd Generation Intel Xeon Scalable processors, compared to TensorFlow on 2nd Generation Intel Xeon Scalable processors.[1] This improvement was achieved in two ways:

- **Hardware:** Upgrading from 2nd Gen to 3rd Gen Intel Xeon Scalable processors brought significant performance improvements due to built-in acceleration for training and inference workloads as well as increased cache architecture and memory bandwidth and channels.
- **Software:** Using Intel performance optimization tools to analyze and tune reference models provided with OpenVINO helped ST Engineering achieve substantial improvements in performance.[1] In addition, they were able to develop a solution more quickly than would otherwise have been possible.

These results show the potential for developers of smart city solutions who leverage Intel technology to provide a highly cost-effective path to VA solutions—with the flexibility to optimize and scale deployment across whatever computing resources are available, whether in the cloud, at the edge, or both.

Learn more

ST Engineering's Video Analytics Platform ›

Intel Distribution of OpenVINO toolkit ›

3rd Generation Intel Xeon Scalable Processors ›

Intel Deep Learning Boost ›

intel® + ST Engineering

1. Testing conducted on ubuntu 20_04 comparing 2nd Generation Intel® Xeon® Gold 6230 using OpenVINO™ 2021.3.394 CNN model optimizations (TensorFlow fp32 vs. OpenVINO fp32 vs. OpenVINO int8) to 3rd Generation Intel® Xeon® Gold 6338 with OpenVINO 2021.3.394 CNN model optimizations (OpenVINO int8). Testing done by Intel in September 2021.

Detailed testing configuration information for 2nd Generation Intel Xeon Scalable processors

Hardware	Item	Description
CPU	Platform	Intel Purley CRB E63448-400
	Product	Intel® Xeon® Gold 6230
	Number of Cores	20 x 2 sockets
Memory	Type	DDR4
	Size	192 GiB
	Channels	12
	Speed	2666 MHz
BIOS	Version	PLYXCRB1.86B.0556.D04.1810190259
	Microcode	0x4003102
Software	Item	Description
Compiler Version	GCC Version	7.5.0
	Linker Version	8.28
Operating System	OS Version	Ubuntu_20_04
	Kernel Version	5.8.0-63-generic
	Docker Version	20.10.7
	Docker Operating System	Ubuntu_18_04_5 LTS
	Docker Kernel Version	5.8.0-63-generic
Benchmark Software	OpenVINO Version	OpenVINO 2021.3.394
	GStreamer Version	1.16.2
	STEEng Video Pipeline	V2.0
CNN Models	OpenVINO Model Zoo	Person-vehicle-bike-detection-2000 fp32 Person-vehicle-bike-detection-2000 int8
	TensorFlow	SSD-MobileNetV2-Coco fp32

Detailed configuration information for 3rd Generation Intel Xeon Scalable processors::

Hardware	Item	Description
CPU	Platform	Intel Ice Lake
	Product	Intel® Xeon® Gold 6338
	Number of Cores	32 x 2 sockets
Memory	Type	DDR4
	Size	256 GiB (16 GB x 16)
	Channels	16
	Speed	3200 MHz
BIOS	Version	1.2.4
	Microcode	0xd0002b1
Software	Item	Description
Compiler Version	GCC Version	7.5.0
	Linker Version	8.28
Operating System	OS Version	Ubuntu_20_04
	Kernel Version	5.4.0-84-generic
	Docker Version	20.10.7
	Docker Operating System	Ubuntu_18_04_5 LTS
	Docker Kernel Version	5.4.0-84-generic
Benchmark Software	OpenVINO™ Version	OpenVINO 2021.3.394
	GStreamer Version	1.16.2
	STEEng Video Pipeline	V2.0
CNN Models	OpenVINO Model Zoo	Person-vehicle-bike-detection-2000 int8
	TensorFlow	SSD-MobileNetV2-Coco fp32

2. Ezzat, M. A.; Abd El Ghany, M. A.; Almotairi, S.; Salem, M. A.-M. “Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends.” Sensors, 2021, Vol. 21, Issue 9. https://doi.org/10.3390/s21093222

3. Cost data determined using dell.com quotes for customized configurations in September of 2021. PowerEdge R740xd2 Rack Server with Intel® Xeon® Gold 6230 costing USD 14,237.06 vs. PowerEdge R750 Rack Server with Intel® Xeon® Gold 6338 costing USD 17,775.95.

Notices and disclaimers

ST Engineering does not own nor have access to data as the developer. It is the responsibility of the customer to comply with the laws and regulations of the relevant jurisdictions.

Performance varies by use, configuration, and other factors. Learn more at intel.com/PerformanceIndex.

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.

Intel® Advanced Vector Extensions (Intel® AVX) provides higher throughput to certain processor operations. Due to varying processor power characteristics, using AVX instructions may cause, a) some parts to operate at less than the rated frequency and, b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration, and you can learn more at intel.com/go/turbo.

Intel is committed to respecting human rights and avoiding complicity in human rights abuses. See Intel's Global Human Rights Principles. Intel® products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.

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.

Your costs and results may vary.

0522/ADS/CMD/PDF