A Principled Technologies report: Hands-on testing. Real-world results.

Support your modern distributed microservices applications using VMware Tanzu Service Mesh on servers enabled by 3rd Generation Intel Xeon Scalable processors

VMware TSM offers automation, workflows, and technologies that optimize service mesh operations and performance

Software developers are increasingly using microservices Kubernetes architectures, where applications comprise a variety of independent services. A service mesh is a platform that coordinates communication and security between these services.[1]

We conducted hands-on testing to explore how the automation, workflows, and technologies in VMware® Tanzu Service Mesh™ (TSM) optimize service mesh operations and performance. We first deployed a microservices application, distributed over two Kubernetes clusters, with secure inter-cluster communications for the services. We did this twice, once using TSM and once using only Istio. (Note that TSM deploys a version of Istio onto the Kubernetes client cluster, where application workloads run.) We also tested two performance-optimization strategies: a TLS optimization scheme that uses features provided by 3rd Generation Intel® Xeon® Scalable processors and a new TCP bypass scheme.[2,3,4] We measured the TCP performance improvements between services in a TSM deployment by comparing web service performance with and without these optimizations.

We found that using VMware Tanzu Service Mesh to deploy our microservices service mesh environment required much less time and many fewer steps than using the native Istio distribution. We also found that for communications between pods running on the same node in the TSM environment, using the TLS handshake acceleration lowered request duration by up to 47.1 percent while nearly doubling performance in terms of queries per second and using this TCP bypass optimization lowered request duration by up to 11.4 percent.

74% less time
33% fewer steps

To deploy a service mesh environment using VMware TSM vs. using only Istio

1.9x the performance
Up to 47.1% lower latency

with Intel multi-buffer cryptography on 3rd Generation Xeon Scalable processors vs. without the optimization

Up to
11.4% lower latency

with TCP bypass vs. without the optimization

Kubernetes microservices and the need for service mesh

Across a wide range of industries, companies are developing software using a Kubernetes microservices architecture, with multiple independent services making up applications.[5] In this building-block approach, a team developing an ecommerce application could use one web server with a back-end database, but that application could comprise services such as credit card verification, product lists, inventory, advertisements, shopping cart, newsletter signup, and so on. This approach gives developers flexibility, letting them use different programming languages, frameworks, and databases for these services, and makes it easier to test new components. A microservices approach also introduces complexity, however; communications between the services must be secure, and the encryption that is necessary for security can increase latency.

To address these concerns, many environments use a service mesh, which VMware describes as "a modern connectivity and security run-time platform" that handles service-to-service communication and security, monitoring distributed tracing, and resiliency.[6]

About VMware Tanzu Service Mesh

According to VMware, Tanzu Service Mesh "deploys a curated version of Istio [open source service mesh]"[7] and "elevates it to more of a distributed application framework that extends far beyond service-to-service communications and provides advanced security capabilities, resiliency, and automated operations for the application — regardless of which clouds its services are running on."[8]

Tanzu Service Mesh offers "advanced, end-to-end connectivity, security, and insights for modern applications—across application end-users, microservices, APIs, and data—enabling compliance with Service Level Objectives (SLOs) and data protection and privacy regulations."[9]

Learn more at **https://tanzu.vmware.com/service-mesh**.

How do VMware Tanzu Service Mesh and Istio relate?

When you onboard a client Kubernetes cluster into the Tanzu Service Mesh SaaS solution, it deploys a managed and curated version of Istio onto the Kubernetes client cluster, where application workloads run.[10]

According to VMware, "The Tanzu Service Mesh Global Controller uses Istio for certain local control capabilities while also managing the life cycle of that Istio deployment. Customers can choose to utilize this Istio deployment directly or utilize Tanzu Service Mesh's application programming interface (API) and create a global namespace… which provides automated Istio operations and adds additional layers of policy."[11]

VMware states that the TSM global namespace offers the following advanced zero-trust security and compliance capabilities, which are unique to TSM: end-to-end mTLS encryption from service to service without regard to cluster, site, or cloud; access policies for micro-segmentation at the application level; API control and segmentation; PII tracking and data leakage protection for personally identifiable information; east-west threat detection.[12]

Overview of our testing

To explore the ways that the automation and workflows in VMware Tanzu Service Mesh simplify service mesh operations, we deployed one microservices application over two bare-metal Kubernetes clusters with secure inter-cluster communications via Mutual Transport Layer Security (mTLS). We began with two separate bare-metal Kubernetes clusters of four nodes each, using eight Dell™ PowerEdge™ R650 servers powered by 3rd Generation Intel Xeon Scalable processors. (See Figure 1.)

We recorded the number of steps and the amount of time required to deploy the service mesh on each cluster, connect the two clusters, deploy the application, and configure secure, mutually trusted, encrypted communication between services running on different clusters. We did this twice, once using TSM and once using Istio alone.

To explore the ways that the technologies developed for VMware Tanzu Service Mesh and Istio can improve service mesh performance, we used the TSM environment and the Fortio and k6 load testing tools. We conducted tests to determine whether using the Intel multi-buffer cryptography feature for TLS communications in a service mesh on servers with Intel 3rd Generation Scalable processors could improve performance. To do so, we simplified the mesh and the application, redeploying TSM on only one four-node cluster and creating encrypted web traffic between two services: a k6 web client and a Fortio web server. Second, we used a similar approach to measure the performance improvement that a TCP bypass optimization could provide, though we used a Fortio web client to generate load on the server.

For complete details on both our hardware configurations and test procedures, see the science behind the report.

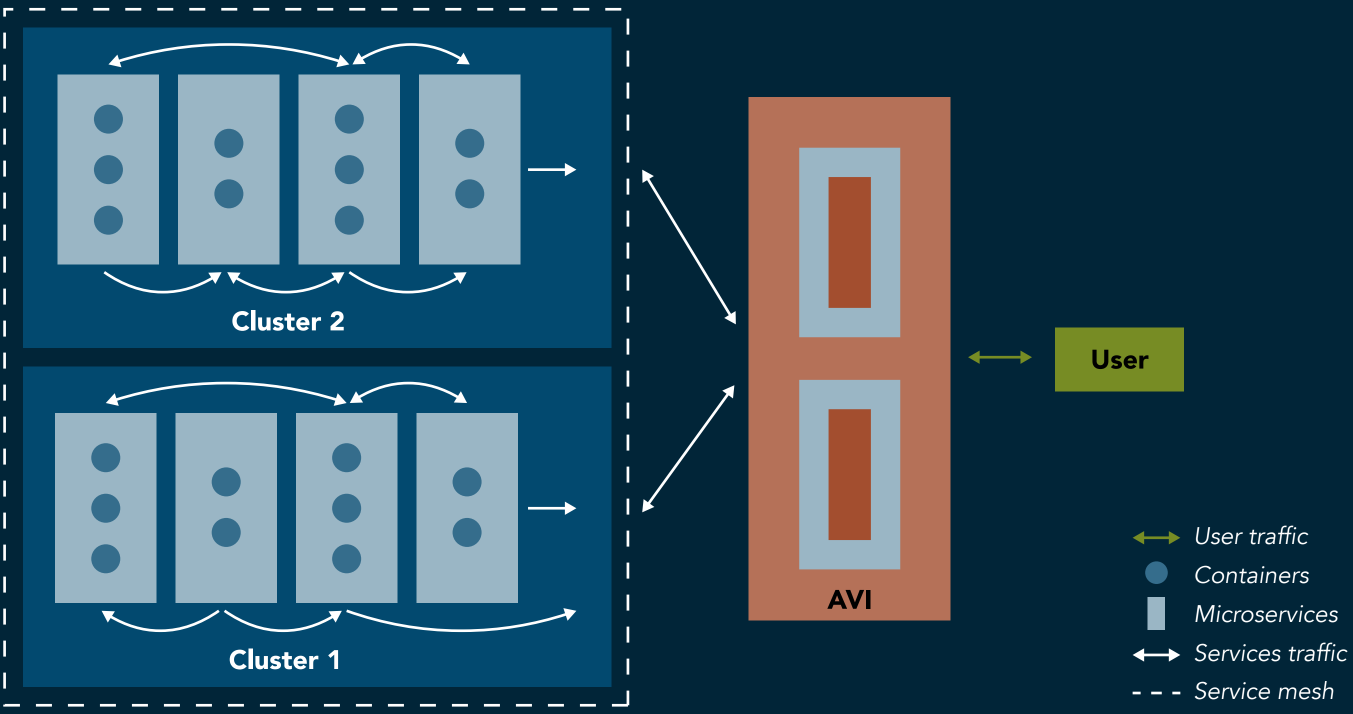


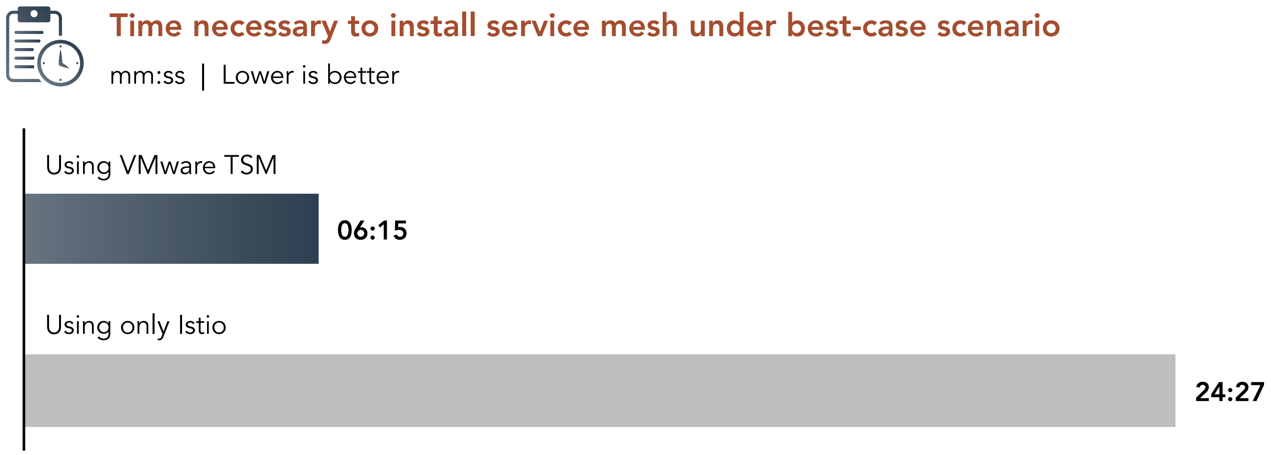
Figure 1: Diagram of our test environment. Source: Principled Technologies.

How did the automation and workflows in TSM reduce the time to create a multi-cluster service mesh and deploy a microservices application on it?

To quantify the effort- and time-saving value of the automation and workflows in TSM, we selected two in-house engineers and had them record the time they needed to create a multi-cluster service mesh and deploy a microservices application on it. They performed this scenario twice, once using TSM and once using only Istio. The engineers used a demo application from Google that implements an Online Boutique in microservices.[13] They used a standard starting point of two pre-existing Kubernetes clusters with an AVI load balancer configured and operating within the two clusters, and recorded the time and steps they needed to deploy a microservices application on each service mesh spanning two clusters with secure mTLS communication between all services.

Engineer 1 began the deployment familiar with the concepts, but with no personal experience using either TSM or Istio. As he worked and resolved issues, he took detailed notes and created step-by-step instructions that Engineer 2 needed only to execute. This approach gave us insight into both typical and best-case scenarios for deployment speed using TSM and using only Istio.

Engineer 1 needed roughly 30 minutes to execute the scenario using VMware Tanzu Service Mesh and roughly 3 hours to do so using only Istio. These approximations include the necessary and realistic time Engineer 1 spent performing first-time research. When Engineer 2 used the detailed instructions that Engineer 1 had created, he needed 6 minutes and 15 seconds to complete the scenario with VMware Tanzu Service Mesh and 24 minutes and 27 seconds to do so using only Istio. Using VMware TSM reduced the time by 74 percent. Figure 2 shows the time that Engineer 2 required, representing a best-case scenario where the engineer doing the work has detailed instructions.


Time necessary to install service mesh under best-case scenario

mm:ss | Lower is better

Figure 2: Time an engineer needed to install service mesh when working with detailed instructions. Less time is better. Source: Principled Technologies.

To understand why completing the exercise using VMware TSM saved so much time, let's look at the steps involved. In this section, we present a high-level overview of the time and steps involved in carrying out our scenario with and without TSM, starting with a fresh bare-metal Kubernetes deployment with two clusters. (Note, in the science behind the report, we provide the detailed step-by-step directions that Engineer 1 prepared for Engineer 2.)

Table 1 presents the five basic tasks our engineers completed when executing our test scenario using only Istio. It states the number of steps and amount of time each step required for Engineer 2, who followed the detailed instructions that Engineer 1 had prepared.

Table 1: The tasks, number of steps, and amount of time involved in executing our multicluster test scenario using only Istio. Source: Principled Technologies.

Tasks using only Istio	Number of steps	Time in mm:ss
Install Istio mesh on the clusters and deploy the Online Boutique application	23	10:03
Install Multi-Primary on different networks	13	10:48
Make application-specific modifications to Istio's default settings	1	02:31
Deploy the Online Boutique application to the mesh	6	00:45
Verify the Online Boutique application works in the two-cluster Istio service mesh	2	00:20
Total	45	24:27

Table 2 presents the four basic tasks our engineers completed when executing our test scenario using TSM. Note that it was not necessary to make any application-specific modifications to the TSM default settings. Like Table 1, this table shows the time that Engineer 2 required to follow the detailed instructions that Engineer 1 had prepared.

Table 2: The tasks, number of steps, and amount of time involved in executing our multicluster test scenario using TSM. Source: Principled Technologies.

Tasks using VMware TSM	Number of steps	Time in mm:ss
Install TSM	15	04:36
Create the Global Namespace	8	00:34
Deploy the Online Boutique application to the mesh	5	00:45
Verify the Online Boutique application works in TSM with two-clusters	2	00:20
Total	30	06:15

Why did VMware TSM save so much time?

Based on the number of steps alone, one might assume that using VMware TSM to complete our scenario would have taken roughly two-thirds as long as using only Istio. In fact, as we noted above, it took just over one-fourth the time, a savings of 74 percent. We attribute this discrepancy to the fact that when using Istio alone, our engineers had to perform a disproportionate number of tasks that our engineers characterized as cumbersome to execute. Many Istio steps required detailed command line work that had to be very precise. In contrast, the TSM deployment process was largely automated. Our engineers reported that the automation and workflow capabilities simplified many activities, making them quick and easy to execute.

About 3rd Gen Intel Xeon Scalable processors

According to Intel, 3rd Gen Intel Xeon Scalable processors are "[o]ptimized for cloud, enterprise, HPC, network, security, and IoT workloads with 8 to 40 powerful cores and a wide range of frequency, feature, and power levels."[14]

Their features include Intel Advanced Vector Extensions 512 (Intel AVX-512), which Intel says "[b]oosts performance and throughput for the most demanding computational tasks in applications such as modeling and simulation, data analytics and machine learning, data compression, visualization, and digital content creation."[15]

To learn more about the 3rd Generation Intel Xeon Scalable processor family, visit https://www.intel.com/content/www/us/en/products/docs/processors/xeon/3rd-gen-xeon-scalable-processors-brief.html.

How did these acceleration technologies improve service mesh performance?

VMware Tanzu Service Mesh, like Istio, adds sidecar proxies to seamlessly connect microservices. That general approach permits opportunities for increasing performance with TSM or Istio in certain configurations. For example, TSM can use mTLS to secure communications between microservices and the mesh components that connect them. The use of TLS is ubiquitous and any speedups in TLS could benefit mesh performance.

The slowest part of the TLS algorithm is the initial stage where the two ends establish trust and exchange cryptographic keys using the RSA or similar algorithms. Intel has written cryptographic libraries that make use of AVX-512 operations in its 3rd Generation Intel Xeon Scalable processors. These libraries can potentially accelerate TLS operations in TSM. One approach to speeding TLS with AVX-512 in TSM is Intel multi-buffer cryptography, which uses multiple buffers, processes RSA operations in a SIMD pipeline, potentially enabling greater throughput and reduced latencies.[16] In our TLS environment, we tested this approach to optimizing TLS by having 400 simulated users send TLS-secured communications to one web server for 4 minutes. Because the k6 user dropped the channel after it sent each message and received the reply, creating a new TLS channel for the next message was necessary, and offered another opportunity for TLS acceleration.

A second opportunity for increasing performance arises when two microservices run on the same node; one can decrease the number of times a network packet flows through the OS's TCP module by having their proxies communicate via an eBPF routine. That routine enforces the security and routing controls without having to use all parts of the SDN machinery. We tested one such eBPF TCP bypass scheme[17] to determine its performance gains over the default TCP stack.

Accelerating TSM's RSA operations with Intel multi-buffer cryptography on Intel 3rd Generation Scalable processors

To quantify the impact of the Intel multi-buffer cryptography optimization on RSA operations, we set up a new four-node Kubernetes cluster and deployed TSM. To focus on the AVX-512 capabilities of these processors, we modified the default TSM configuration slightly so that the Istio pods performing the TLS operations had the same memory and CPU resources as those we used with the Intel multi-buffer cryptography solution. We used the k6 load-generating tool to simulate 400 users sending small web requests to the Fortio server. We used k6 because we wanted greater control over the client-side TLS, and we did not need to send requests at a fixed rate. k6 delivered two metrics: request duration, or latency, and queries per second (QPS). We measured performance with and without the Intel multi-buffer cryptography optimization for 3rd Generation Intel Xeon Scalable processors.

Figure 3 shows our findings for 99th percentile latency. Using Intel multi-buffer cryptography reduced request duration by 47.1 percent.

99th percentile request duration (latency) with and without Intel multi-buffer cryptography enabled Milliseconds | Lower is better

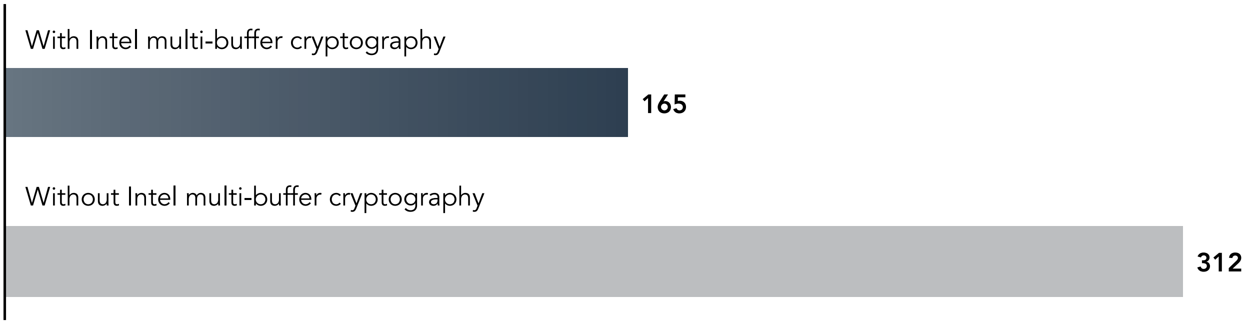


Figure 3: TLS acceleration test. Performance impact of enabling Intel multi-buffer cryptography on TSM using Intel 3rd Generation Scalable processors. Lower latency is better. Source: Principled Technologies.

Figure 4 shows our findings for performance in terms of queries per second. Using Intel multi-buffer cryptography nearly doubled performance, achieving 1.9 times as many queries per second.

Queries per second with and without Intel multi-buffer cryptography enabled
Queries per second | Higher is better

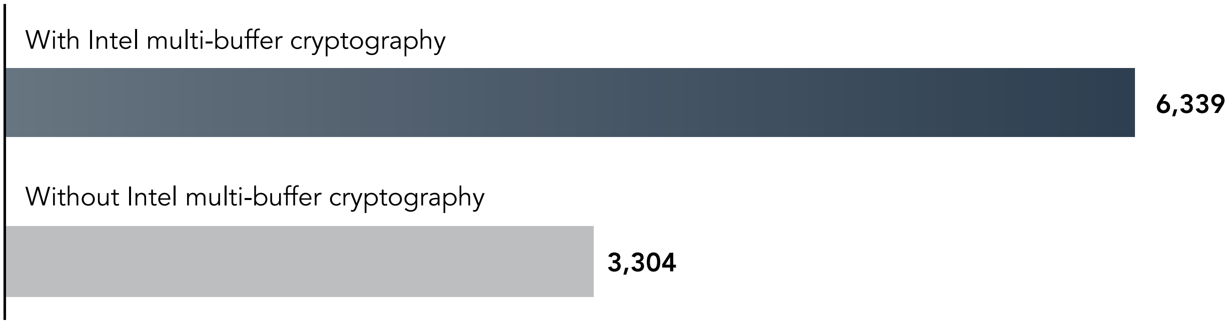


Figure 4: TLS acceleration test. Performance impact of enabling Intel multi-buffer cryptography on TSM using Intel 3rd Generation Scalable processors. Greater QPS is better. Source: Principled Technologies.

Accelerating intra-node communications with a TCP bypass strategy

To quantify the impact of the TCP bypass optimizations on intranode communications, we set up a four-node Kubernetes cluster and deployed TSM. We used a Fortio client pod to send 1,000 1KB requests per second for various numbers of virtual users to a Fortio web server pod. The Fortio client measured the request duration, or latency. We measured performance with and without the optimization.

Figure 5 shows our findings for 99th percentile latency across the four virtual-user counts we tested with and without the TCP bypass optimization. Using this optimization consistently reduced request duration, with improvements ranging from 5.0 percent at 16 virtual users to 11.4 percent at 32 virtual users.

99th percentile request duration (latency) with and without TCP bypass enabled

Lower is better

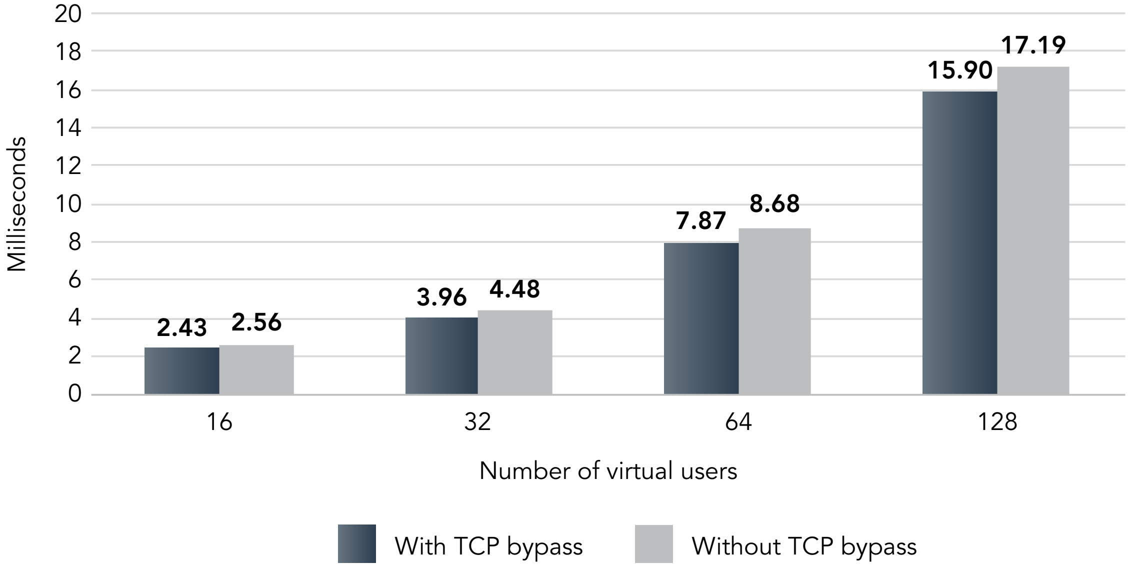


Figure 5: TCP data-path optimization test. Performance impact of enabling TCP bypass at various thread counts. Lower latency is better. Source: Principled Technologies.

Additional features of VMware Tanzu Service Mesh

According to VMware, a default installation of TSM offers features that support "Deep application visibility and actionable insights."[18] Specifically, "Tanzu Service Mesh helps teams overcome the performance and security visibility gaps resulting from distributed microservices architectures and adoption of multiple platforms and clouds. Operations teams have access to rich troubleshooting tools, including multi-cloud topology maps and traffic flows, performance and health metrics, and application-to-infrastructure correlation. [...] Troubleshooting application issues or investigating security incidents becomes much easier—reducing mean time to identify/repair and detect/respond."[19]

We did not test these features. Our default installation of Istio did not include these features.

Conclusion

If your organization uses a microservices Kubernetes architecture, a service mesh is a valuable tool for coordinating communication and security among services. In our testing, we deployed a microservices application, distributed over two Kubernetes clusters with secure inter-cluster communications for the services. We found that using VMware TSM to carry out this task reduced the amount of time necessary by 74 percent compared to using only Istio. In performance testing of the TSM environment, the TCP bypass optimization reduced request duration by as much as 11.4 percent and the Intel multi-buffer cryptography optimization for Intel 3rd Generation Xeon Scalable processors reduced request duration by up to 47.1 percent while nearly doubling performance.

[1] Niran Even-Chen, Oren Penso, Sergio Pozo, and Susan Wu, "Service Mesh For Dummies, VMware 2nd Special Edition," accessed October 25, 2022, https://tanzu.vmware.com/content/ebooks/service-mesh-for-dummies-2022.

[2] Intel multi-buffer cryptography is part of the Intel Integrated Performance Primitives Cryptography library. To learn more, see https://github.com/intel/ipp-crypto/blob/develop/sources/ippcp/crypto_mb/Readme.md.

[3] Manish Chugtu, "TLS Handshake Acceleration with Tanzu Service Mesh," accessed September 26, 2022, https://blogs.vmware.com/networkvirtualization/2022/08/tls-handshake-acceleration-with-tanzu-service-mesh.html.

[4] Manish Chugtu, "Tanzu Service Mesh Acceleration using eBPF", accessed September 26, 2022, https://blogs.vmware.com/networkvirtualization/2022/08/tanzu-service-mesh-acceleration-using-ebpf.html.

[5] Solo.io, "New Research Reveals Microservices, Service Mesh Critical to Modern Digital Transformation Efforts," accessed September 26, 2022, https://www.globenewswire.com/en/news-release/2022/06/16/2464004/0/en/New-Research-Reveals-Microservices-Service-Mesh-Critical-to-Modern-Digital-Transformation-Efforts.html.

[6] Niran Even-Chen, Oren Penso, Sergio Pozo, and Susan Wu, "Service Mesh For Dummies, VMware 2nd Special Edition," accessed October 25, 2022, https://tanzu.vmware.com/content/ebooks/service-mesh-for-dummies-2022.

[7] VMware, "Top Use Cases for VMware Tanzu Service Mesh, Built on VMware NSX," accessed October 25, 2022, https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/nsx/vmware-tanzu-usecases.pdf.

[8] Niran Even-Chen, Oren Penso, Sergio Pozo, and Susan Wu, "Service Mesh For Dummies, VMware 2nd Special Edition," accessed October 25, 2022, https://tanzu.vmware.com/content/ebooks/service-mesh-for-dummies-2022.

[9] VMware, "VMware Tanzu Service Mesh," accessed September 26, 2022, https://tanzu.vmware.com/service-mesh.

[10] Niran Even-Chen, Oren Penso, Sergio Pozo, and Susan Wu, "Service Mesh For Dummies, VMware 2nd Special Edition," accessed October 25, 2022, https://tanzu.vmware.com/content/ebooks/service-mesh-for-dummies-2022.

[11] Niran Even-Chen, Oren Penso, Sergio Pozo, and Susan Wu, "Service Mesh For Dummies, VMware 2nd Special Edition."

[12] Niran Evenchen, "Using Global Namespaces and Zero-Trust Policies with VMware Tanzu Service Mesh," accessed September 30, 2022, https://tanzu.vmware.com/content/blog/using-global-namespaces-zero-trust-policies-vmware-tanzu-service-mesh.

[13] Github, GoogleCloudPlatform/microservices-demo, accessed September 26, 2022, https://github.com/GoogleCloudPlatform/microservices-demo.

[14] Intel, "3rd Gen Intel® Xeon® Scalable Processors Brief," accessed September 26, 2022, https://www.intel.com/content/www/us/en/products/docs/processors/xeon/3rd-gen-xeon-scalable-processors-brief.html.

[15] Intel, "3rd Gen Intel® Xeon® Scalable Processors Brief."

[16] Manish Chugtu, "TLS Handshake Acceleration with Tanzu Service Mesh," accessed September 26, 2022, https://blogs.vmware.com/networkvirtualization/2022/08/tls-handshake-acceleration-with-tanzu-service-mesh.html.

[17] Manish Chugtu, "Tanzu Service Mesh Acceleration using eBPF", accessed September 26, 2022, https://blogs.vmware.com/networkvirtualization/2022/08/tanzu-service-mesh-acceleration-using-ebpf.html.

[18] VMware, "VMware Tanzu Service Mesh," accessed September 26, 2022, https://tanzu.vmware.com/service-mesh.

[19] VMware, "VMware Tanzu Service Mesh."

Read the science behind this report at https://facts.pt/PSaJ15T ▶

Principled Technologies®

Facts matter.®

This project was commissioned by VMware.