Kubernetes on VMware vSphere vs. bare metal: Which delivered better density and performance?

A Kubernetes cluster on VMware vSphere achieved comparable cloud-based workload performance vs. a bare-metal Kubernetes cluster on the same hardware—and performed better on some tests

Organizations running containerized workloads in Kubernetes® have a number of choices. Our hands-on testing used the same server hardware to compare the cloud performance of two single-server Kubernetes clusters: (1) a virtualized environment using VMware® vSphere® 7 Update 1, Ubuntu Linux®, and a standalone deployment of VMware Tanzu™ Kubernetes Grid (TKG) and (2) a bare-metal cluster running Ubuntu Linux and open-source Kubernetes.

We found that the two clusters delivered comparable performance on two cloud-based workloads. On the TKG cluster, VMware TKG made it easy to get started, and the cluster also supported greater density than the bare-metal cluster. These results make VMware vSphere a very attractive option for users containerizing workloads with Kubernetes.

Comparable—and sometimes better—performance*

The flexibility and efficient resource utilization of virtualization

Support for greater pod density*

**vs. bare-metal cluster*

About VMware Tanzu Kubernetes Grid

VMware describes Tanzu Kubernetes Grid as "a CNCF-certified, enterprise-ready Kubernetes runtime that streamlines operations across a multi-cloud infrastructure," and states that it offers simplified instruction, automated multi-cluster operation, integrated platform services, and open source alignment.[1]

Learn more at https://tanzu.vmware.com/kubernetes-grid.

The benefits of virtualization without performance loss

You might assume that running Kubernetes on VMware vSphere with TKG would force you to give up a certain amount of performance in exchange for the convenience and flexibility that come with virtualization.

To put that assumption to the test, we ran two compute-intensive cloud workloads on these single-server Kubernetes cluster configurations using the same server hardware, a two-socket Dell EMC™ PowerEdge™ R740xd with Intel Xeon Platinum 8164 26-core processors (for a total of 52 cores), 384 GB of 2,400MHz RAM, and two 1.92TB 12Gbps SAS SSDs:

- a virtualized environment using VMware vSphere 7 Update 1 running Ubuntu Linux and TKG
- a bare-metal cluster running Ubuntu Linux and open-source Kubernetes

We also conducted density tests to measure the maximum number of simple pods that each solution could run simultaneously without error. (See the science behind the report for details on our test environment and procedures.)

We found that in addition to being easy to implement, Kubernetes running on vSphere achieved performance comparable to—and, in some cases, better than—that of bare-metal Kubernetes and supported greater density.

About VMware vSphere 7 Update 1

According to VMware, vSphere 7 Update 1 (or vSphere 7U1) offers the following:

- enhanced vSphere Lifecycle Manager hardware compatibility pre-checks for vSAN environments,
- increased number of vSphere Lifecycle Manager concurrent operations on clusters
- vSphere Lifecycle Manager support for coordinated updates between availability zones
- extended list of supported Red Hat Enterprise Linux and Ubuntu versions for the VMware vSphere Update Manager Download Service (UMDS)
- improved control of VMware Tools time synchronization
- increased Support for Multi-Processor Fault Tolerance (SMP-FT) maximums
- virtual hardware version 18
- increased resource maximums for virtual machines and performance enhancements.[2]

Learn more at https://docs.vmware.com/en/VMware-vSphere/7.0/rn/vsphere-esxi-701-release-notes.html.

About CloudXPRT

According to the CloudXPRT website, "CloudXPRT is a cloud benchmark that can accurately measure the performance of modern, cloud-first applications deployed on modern infrastructure as a service (IaaS) platforms, whether those platforms are paired with on-premises (datacenter), private cloud, or public cloud deployments. Regardless of where clouds reside, applications are increasingly using them in latency-critical, highly available, and high-compute scenarios."[3]

Learn more about CloudXPRT at http://cloudxprt.com.

Measuring performance with CloudXPRT

To measure the performance of our two Kubernetes clusters, we used the CloudXPRT benchmark. CloudXPRT consists of a data analytics workload and a web microservices workload, both of which measure application performance on infrastructure as a service (IaaS) platforms. For both workloads, we tested a single configuration on the bare-metal cluster and multiple VM configurations—that is, a variety of counts and sizes—on the VMware TKG cluster.

The CloudXPRT data analytics workload

The CloudXPRT data analytics workload classifies a dataset using the XGBoost gradient-boosting technique. The workload reveals an IaaS stack's ability to optimize and speed training with this model. It makes use of "Kubernetes, Docker, object storage, message pipeline, and monitorization components to mimic an end-to-end IaaS scenario."[4] To better simulate data center activity, the workload creates XGBoost jobs at random times according to a Poisson distribution with an adjustable parameter that determines the average delay between jobs.

The data analytics workload lets testers change the stress on the cluster under test in two ways. The first is using the "burstiness" parameter, which is the average time between submission of data-analytics jobs. The second is using the "CPUs per pod" parameter, which determines the pod's Kubernetes CPU limit and CPU request, the maximum number of OpenMP threads available to the machine-learning process that runs in the pod, and the number of pods that can run on the worker node. The CloudXPRT developers have determined through experimentation that optimizing pod size is important for good workload performance. The workload delivers multiple results per configuration. In this section, we present the best results of tests that met the CloudXPRT criterion that runs have a 95th percentile response time below 90 seconds.

Table 1 shows two of the best CloudXPRT data analytics workload results of the VMware TKG cluster running Kubernetes on the HIGGS1-M machine learning dataset with 100 jobs.

Table 1: Two of the best CloudXPRT data analytics workload results for the VMware TKG cluster. Higher throughput is better. Source: Principled Technologies.

VM configuration		Pod configuration		Throughput (jobs per minute)
No. of nodes	No. of vCPUS per node	No. of pods	No. of CPUs per pod	
3	26	3	20	1.95
1	52	1	48	1.20

Table 2 shows two of the best CloudXPRT data analytics workload results of the bare-metal cluster running Kubernetes on the HIGGS1-M machine learning dataset with 100 jobs.

Table 2: Two of the best CloudXPRT data analytics workload results for the bare-metal cluster. Higher throughput is better. Source: Principled Technologies.

No. of pods	No. of CPUs per pod	Throughput (jobs/min)
3	26	1.96
1	52	1.20

Figure 1 compares the throughput results from Tables 1 and 2. As it shows, the cluster with 26 vCPUs and three VMs, which delivered the best throughput, achieved performance within one-half of a percentage point of that of the higher-performing bare-metal configuration. The lower-performing configurations also performed similarly, with the 52 CPU/vCPU configurations achieving 1.20 jobs per minute for both the TKG and bare-metal clusters.

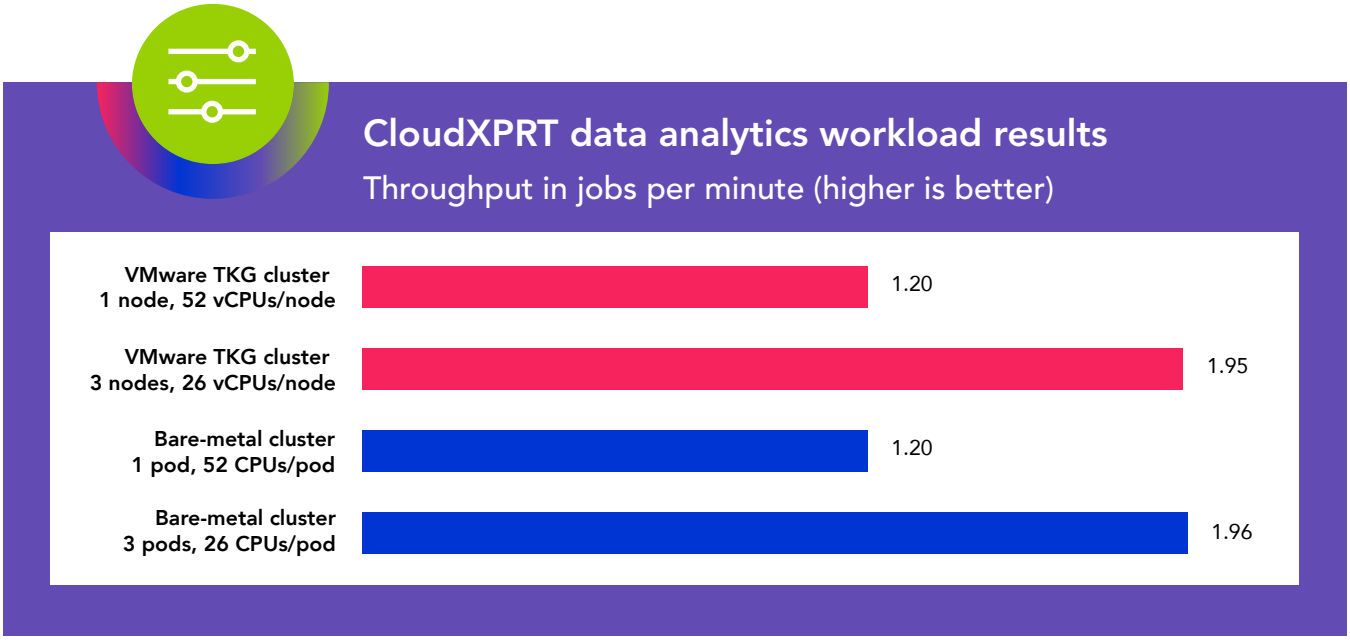
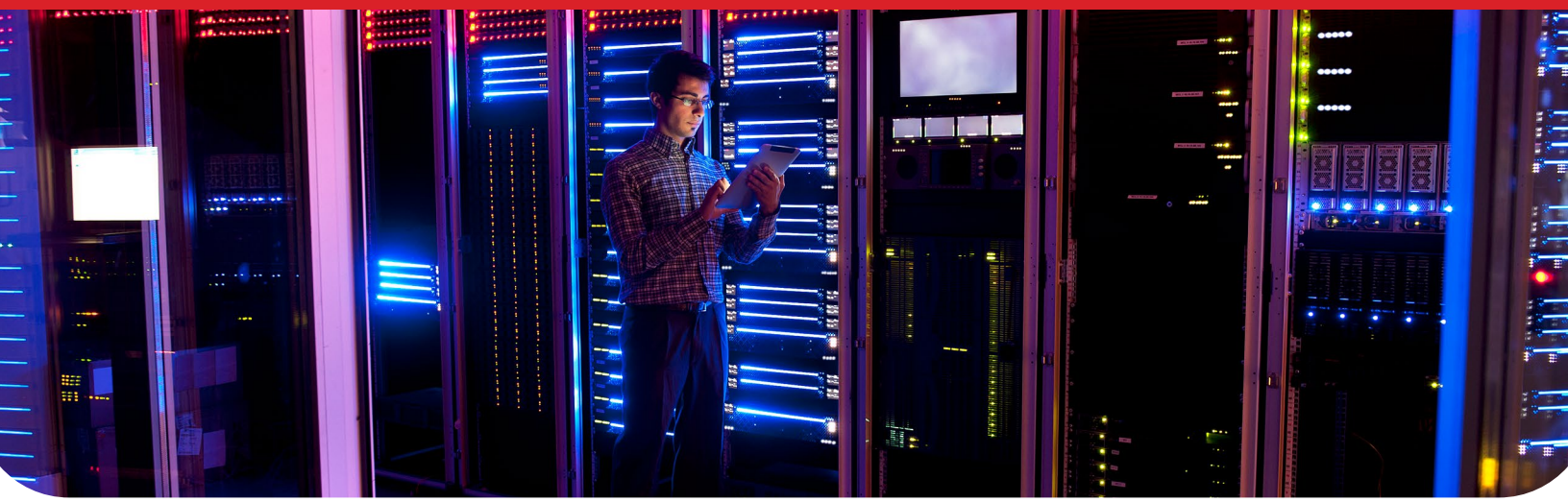


Figure 1: CloudXPRT data analytics workload throughput results for the two clusters. Higher is better. Source: Principled Technologies.

We found that the highest performance for this workload on both testbeds occurred when the number of CPUs/vCPUs available to the worker pods was large, but left enough CPU resources for headroom for Kubernetes system processes, and had the potential to run each pod or VM on one NUMA node. We did not use CPU affinity or pinning, but allowed the native schedulers to assign CPUs/vCPUS to each pod.



The CloudXPRT web microservices workload

In the CloudXPRT web microservices workload, a web application selects stocks, performs Monte Carlo simulations on them, and shows a simulated user options. The workload simulates an end-to-end IaaS scenario using Kubernetes, Docker, NGNIX, REDIS, Cassandra, and monitoring modules.[5] Rather than providing a single score, the benchmark delivers both throughput and latency results; each user's specific needs determine which of these two metrics—or a particular balance between the two—equals the optimal score.[6]

The bare-metal cluster had a single IaaS configuration, and we experimented with a variety of VM counts and sizes for the virtualized testbed. Each VM had a 20GB disk volume and 10 GB of RAM. In contrast to the data analytics workload, which has two tunable parameters, the CloudXPRT web microservices workload cycles through a set of parameters to determine the combination that produces the best throughput and latency results. In this section, we present the results of tests that met a service-level agreement (SLA) criterion of 3 seconds.

Table 3 shows four of the best CloudXPRT web microservices workload results of the VMware TKG cluster running Kubernetes.

Table 3: Four of the best CloudXPRT web microservices workload results for the VMware TKG cluster. Higher throughput rates and lower latencies are better. Source: Principled Technologies.

No. of vCPUs	No. of VMs	Throughput (requests per minute)	Latency (ms)	Notes
52	1	1,367	720	Best combination: second-highest throughput, lowest latency
20	4	1,266	2,249	Comparable throughput performance
26	3	1,258	1,992	Comparable throughput performance
48	2	1,391	1,710	Highest throughput, but also higher latency
32	3	1,341	1,549	Third-highest throughput, and lower latency than the run with the highest throughput

Table 4 shows the best CloudXPRT web microservices workload result of the bare-metal cluster running Kubernetes, 1,263 requests per minute with a latency of 820 milliseconds.

Table 4: The best CloudXPRT web microservices workload result for the bare-metal cluster. Higher throughput rates and lower latencies are better. Source: Principled Technologies.

Throughput	Latency (ms)
1,263	820

Figure 2 compares the results in Tables 3 and 4. As it shows, the cluster with 52 vCPUs and one VM, which delivered the best combination of throughput and latency, outperformed the bare-metal cluster by 8 percent and achieved 12 percent lower latency than the bare-metal cluster. The 48-vCPU and 2-VM configuration performed at an even higher throughput (an increase of 10 percent), though the latency was significantly higher.

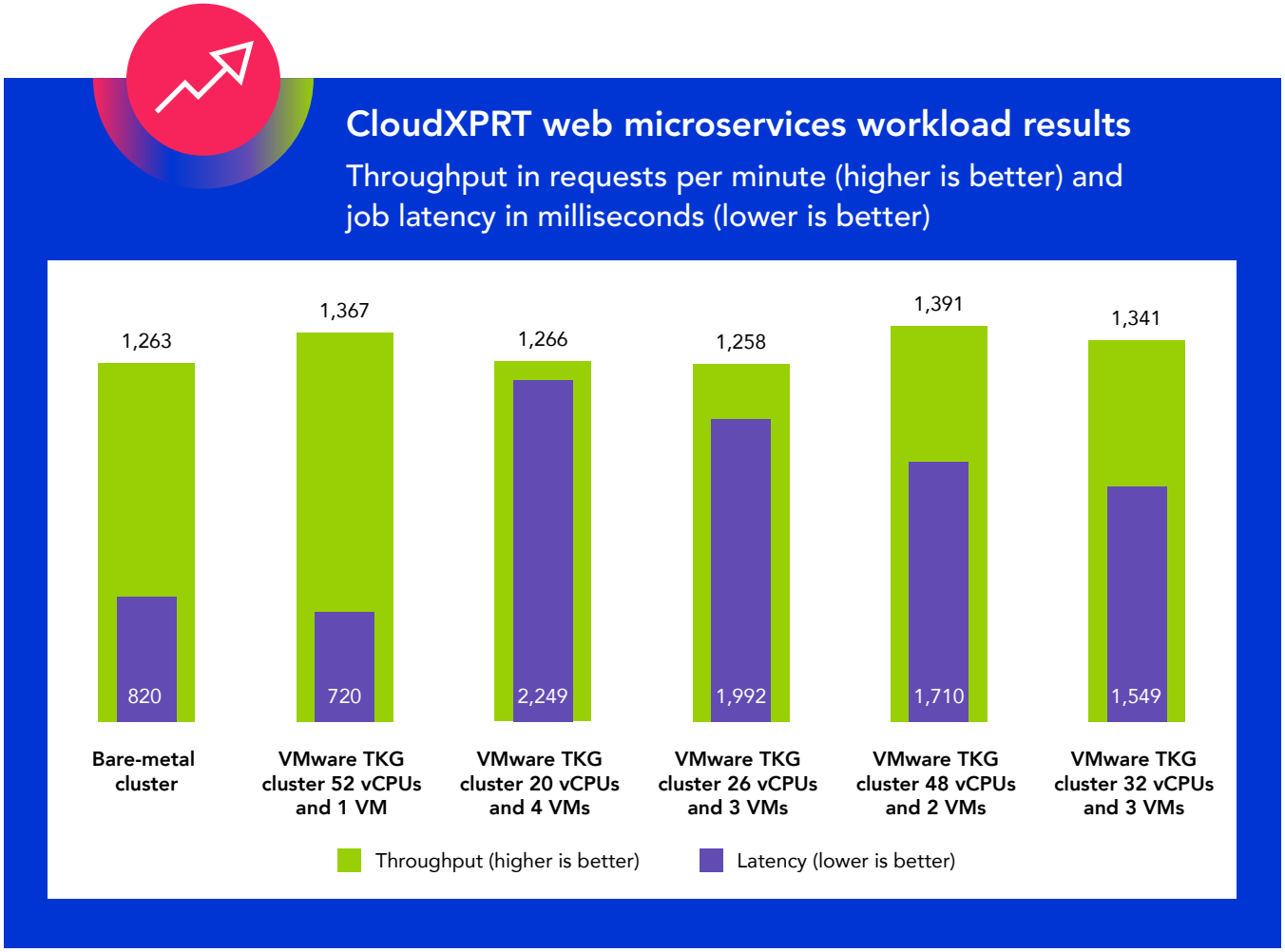


Figure 2: CloudXPRT web microservices workload results for the two clusters. Higher throughput rates and lower latencies are better. Source: Principled Technologies.

Measuring worker node and pod density

By default, Kubernetes allows a maximum of 110 pods per node. This means that a VMware TKG cluster could potentially support many more pods—110 per VM node—than our bare-metal single-server cluster, which could support only 110 pods. We experimented with scaling VM worker nodes using the default Kubernetes maximum of 110 pods to see how many worker nodes and pods the TKG cluster could support. To focus on the Kubernetes systems' ability to handle many pods, we replaced the CPU-heavy CloudXPRT workloads with a minimal web-service workload, which we describe in the science behind the report. We also experimented with the bare-metal Kubernetes cluster, going beyond the 110-pod limit to determine how many pods it could support.

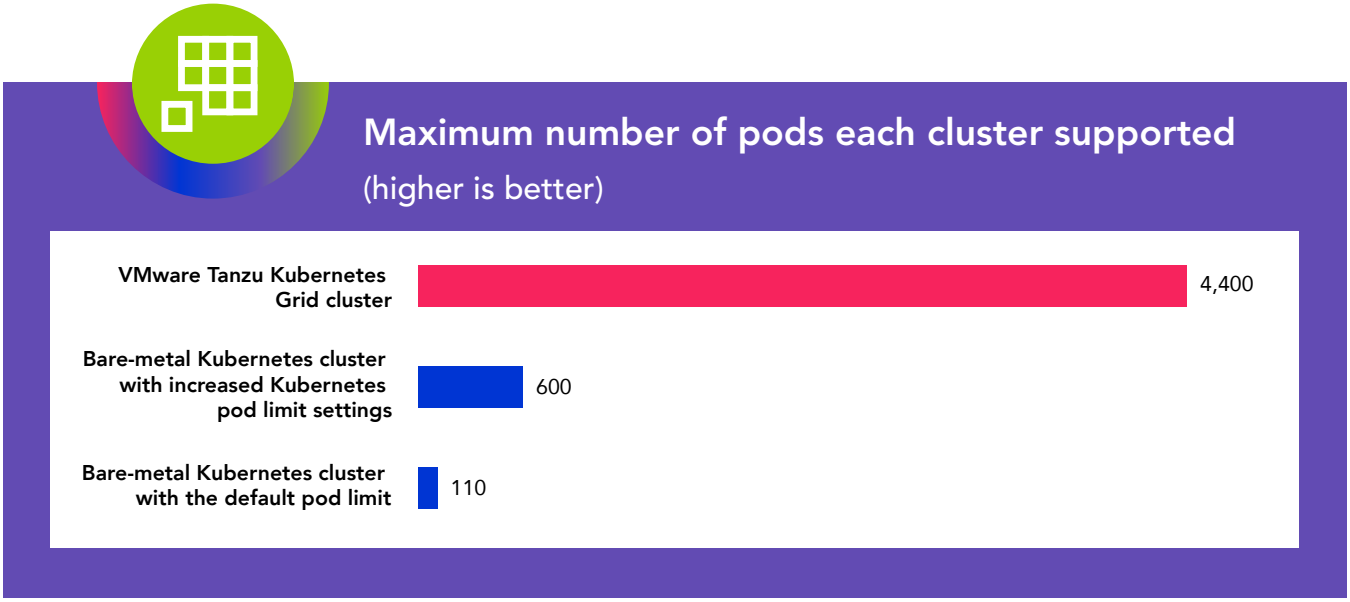


Figure 3: Maximum number of pods the two clusters supported while running a minimal web-service workload. Higher numbers are better. Source: Principled Technologies.

We raised the limit on the bare-metal Kubernetes cluster to allow a maximum of 1,500 pods per node. We then attempted to scale the bare-metal cluster starting with 100 pods, then increasing first by 100 pods and then by 50 pods at a time until reaching the peak number of functioning pods. We observed normal performance, though with increasing warnings of stress on the API server and network pods, only until we had deployed a total of 600 pods. At that point, one calico network pod crashed. In one run, the pod restarted; in two others, it did not.

To determine the maximum number of pods the TKG cluster could support, we scaled the worker node count. For the worker node size, we used two vCPUs with 8GB RAM. We then scaled with the following per-cluster worker VM counts: 1, 25, 30, 35, 40, and 45.

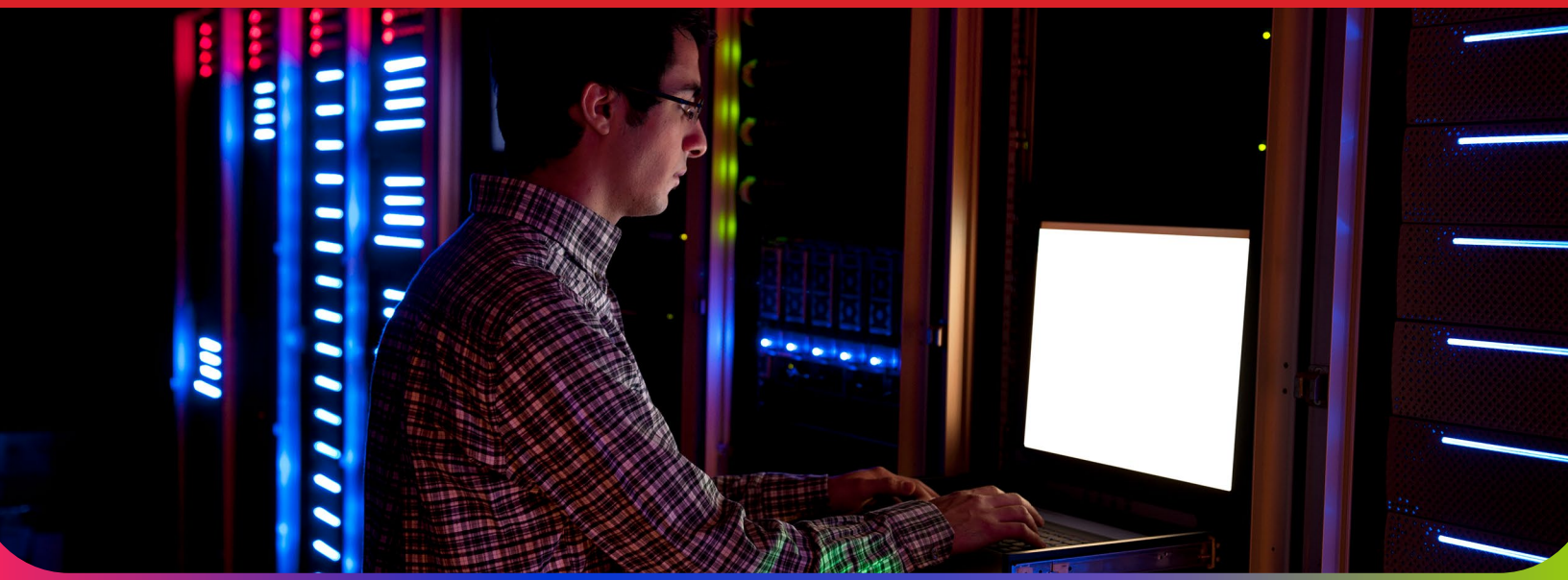
The VMware TKG cluster supported up to 40x the density of the bare-metal cluster with the Kubernetes default settings and up to 7x the density of the bare-metal cluster when we increased the pod limit.

At each worker VM count up to and including the 40-VM one, each worker node was fully populated with the default Kubernetes maximum of 110 pods (107 application + 3 system). For the 40-VM cluster, we deployed 107*40 = 4,280 application pods (4,400 pods total). At this size, the application performed well, but response time increased to the still acceptable 95th-percentile rate of 206 milliseconds. For the 25-VM cluster, the 95th-percentile response time was 38 milliseconds.

We could not fully populate the 45-VM cluster due to the CPU load on the control-plane node, which resulted in a sustained loss of connection to the Kubernetes system. Note that we used a "dev+small" worker cluster with one control-plane node as we kept to the original cluster deployed for the CloudXPRT tests. The ESXi server had 50 GiB of free RAM and so was not swapping while CPU usage was under 50 percent. Using a larger control plane configuration could potentially have let the cluster support even greater density.

We conclude that TKG can support a significantly greater density of pods per server than bare-metal Kubernetes because one bare-metal server can host a multi-node TKG cluster. Using simply the default Kubernetes maximums, the TKG could support 40x as many pods as the bare metal cluster could support. Even if one removed the default pod limit on the bare-metal cluster, the TKG cluster could support 7x the bare metal cluster's best effort.

For details on our pod-density testing, see the science behind the report.



Conclusion

Our testing demonstrates that organizations need not hesitate to run Kubernetes containerized workloads in a virtualized VMware environment. On two compute-intensive cloud-based workloads, a single-server environment running Tanzu Kubernetes Grid on VMware vSphere 7 achieved performance comparable to—and, in some tests, better than—that of a bare-metal single-server environment running Ubuntu Linux and open-source Kubernetes.

---

1 "Solution overview: VMware Tanzu Kubernetes Grid," accessed March 4, 2021, https://d1fto35gcfffzn.cloudfront.net/tanzu/tkg/TKG-Solution-Overview.pdf.

2 "VMware ESXi 7.0 Update 1 Release Notes," accessed March 24, 2021, https://docs.vmware.com/en/VMware-vSphere/7.0/rn/vsphere-esxi-701-release-notes.html

3 "CloudXPRT," accessed March 4, 2021, https://www.principledtechnologies.com/benchmarkxprt/cloudxprt/.

4 Principled Technologies, "Overview of the CloudXPRT Data Analytics Workload" accessed March 4, 2021, https://www.principledtechnologies.com/benchmarkxprt/cloudxprt/2021/Overview-CloudXPRT-Data-Analytics-Workload.pdf.

5 Principled Technologies, "Overview of the CloudXPRT Web Microservices Workload," accessed March 4, 2021, https://www.principledtechnologies.com/benchmarkxprt/counter.php?inline=true&redirect=/benchmarkxprt/cloudxprt/2020/Overview-CloudXPRT-Web-Microservices-Workload.pdf.

6 Principled Technologies, "Overview of the CloudXPRT Web Microservices Workload," accessed March 4, 2021, https://www.principledtechnologies.com/benchmarkxprt/counter.php?inline=true&redirect=/benchmarkxprt/cloudxprt/2020/Overview-CloudXPRT-Web-Microservices-Workload.pdf.

Read the science behind this report at http://facts.pt/3Lxfi7z ▶

Principled Technologies®

Facts matter.®

This project was commissioned by VMware.