PERFORMANCE STUDY – AUGUST 2017

PERFORMANCE OF STORAGE POLICY–BASED MANAGEMENT

VMware vCenter Server 6.5

vmware®

Table of Contents

Introduction

Storage Policy Based Management (SPBM) is a storage policy framework that helps administrators match VM workload requirements against storage capabilities. SPBM runs as an independent service in the vCenter Server.

This paper covers SPBM performance and is divided into two sections. In the first section, we cover performance improvements of SPBM in vCenter 6.5 over vCenter 6.0 U3. In the second section, we cover scaling trends of SPBM in vCenter 6.5.

For this study, we looked at SPBM performance in four different configurations:

- **Storage policies using VAIO (IOF)**
  vSphere APIs for I/O Filtering (VAIO) is a framework to filter I/Os of a VM workload. Starting with vSphere 6.0 U1, storage vendors can present capabilities of filters (such as caching and replication) using the VAIO framework [1] [2].
- **Storage policies using Tags (TAG)**
  Tags allow administrators to attach metadata to objects in the vSphere inventory to make these objects more sortable and searchable. These tags can be used in storage policies to guide storage placement.
- **Storage policies using vSAN capabilities (VSAN)**
  VMware vSAN™ is the software powering hyper-converged infrastructure solutions. SPBM integrates with vSAN to discover the capabilities the vSAN setup offers and presents the administrators options to author storage policies custom-made for the workload requirement [3].
- **Storage policies using VVol capabilities (VVOL)**
  VMware vSphere® Virtual Volumes™ is a vSphere integration and management framework that virtualizes SAN/NAS arrays, enabling a more efficient operational model that is optimized for virtualized environments and centered on the application instead of the infrastructure. SPBM allows capturing storage service-level requirements (such as capacity, performance, and availability) in the form of storage policies to which virtual machines are associated [4].

For detailed information regarding the setup used in these various configurations, please refer to the Appendix.

Performance Improvements

In this section, we compare the performance of SPBM in the vCenter Server 6.5 release against the previous 6.0U3 release. We cover operations of VM provisioning, querying policy associations, and service restart.

VM Provisioning

VM provisioning operations include actions like creating a VM, cloning a VM, migrating a VM, or editing properties of an existing VM. The performance of VM provisioning operations depends on the configuration of the VM being provisioned. This includes the size and type of the VM disks (for example, thin or thick and eager-zeroed vs. lazy zeroed), the number of virtual disks, and storage policies associated with them, among other factors. During a VM provisioning operation, SPBM is also consulted to fetch the policy to be associated with the VM being provisioned. This could either be the user-selected storage policy or the default storage policy of the datastore.

In this section, we study the performance of provisioning operations with VMs that have a user-defined storage policy associated with them and all their virtual disks, keeping all the other parameters constant.

VM Create

In this experiment, we created a virtual machine with two virtual disks, both associated with a storage policy. We created VMs on a vSphere 6.0U3 environment and then repeated the experiment in a vSphere 6.5 environment of the same scale.

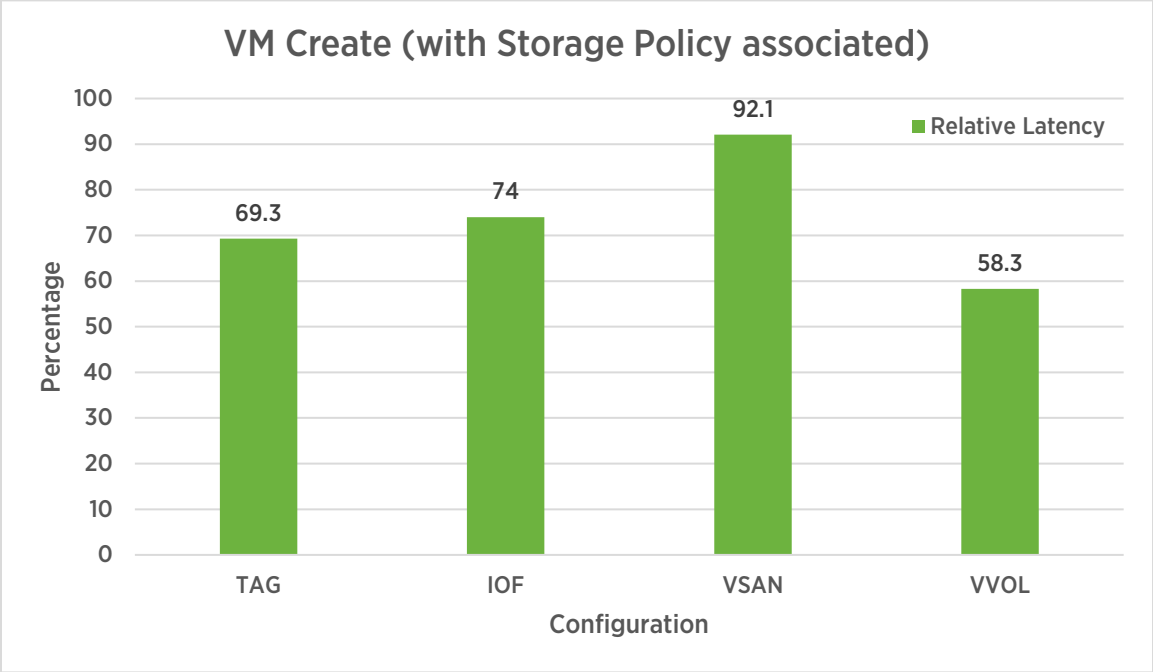

Figure 1. VM create times with storage policies associated shown in vCenter Server 6.5 as a percentage of time taken in vCenter Server 6.0U3 (lower is better)

As shown in Figure 1, vCenter Server 6.5 performs better for VM create operations when compared to 6.0U3. The improvement varies from 8% to 42% across the different configurations.

VM Clone

A storage policy can also be applied at the time of cloning a VM. If not specified, the cloned VM inherits the storage policy of the source VM.

In this experiment, we cloned a VM having two virtual disks, both associated with a storage policy. We cloned VMs on a vSphere 6.0 U3 environment and then repeated the experiment in a vSphere 6.5 environment of the same scale.

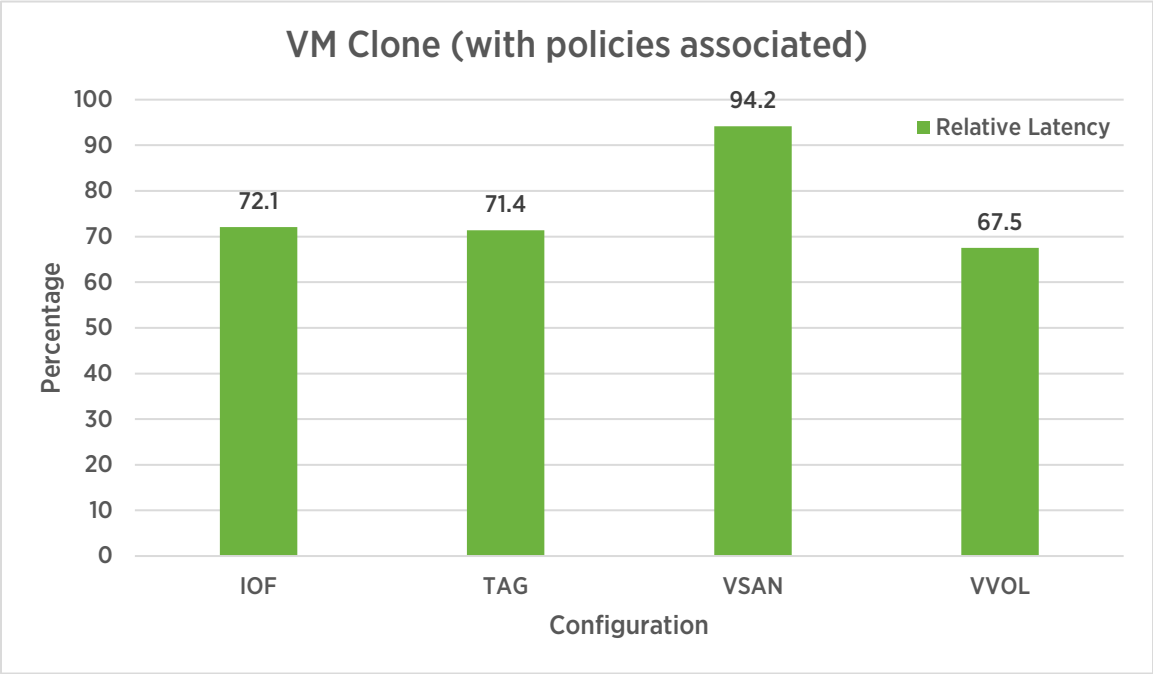

Figure 2. VM Clone with storage policy associated - relative latency of vCenter Server 6.5 as a percentage of vCenter Server 6.0U3 (lower is better)

As we can see, the time taken to clone a VM is less in vCenter Server 6.5 when compared to 6.0 U3. The improvement varies from 6% to 32% across different configurations.

VM Reconfigure

The storage policy associated with a VM/disk can be changed by reconfiguring the VM. Re-applying a storage policy currently associated also triggers a VM reconfigure operation.

In this experiment, we carried out VM reconfigure operations that changed the policy of a VM and its two disks.

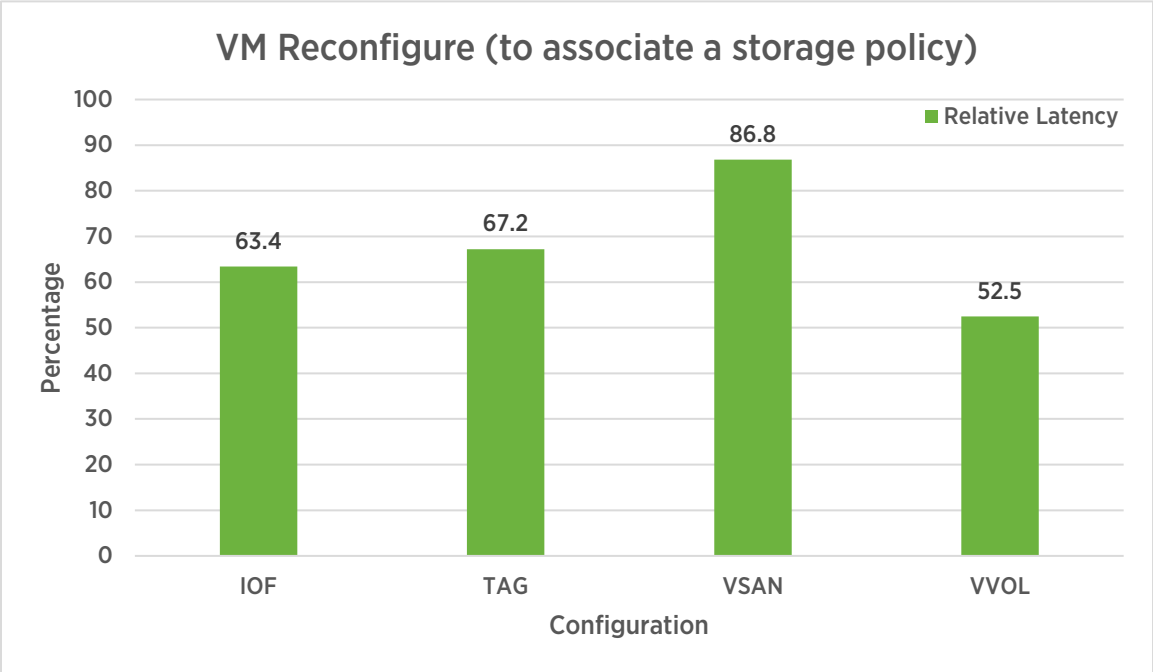

Figure 3. VM reconfigure times when associating a storage policy – latency in vCenter Server 6.5 as a percentage of vCenter Server 6.0U3 (lower is better)

As we can see, reconfiguring a VM is faster in vCenter Server 6.5 when compared to 6.0 U3. The improvement varies from 13% to 47% across the different configurations.

Query Associated Profile

SPBM can be queried for the storage policy associated with a VM/disk. This is often invoked by the vSphere Client as well as by Storage Distributed Resource Scheduler (SDRS) when recommending initial placement and later load balancing workloads in a storage pod.

In this experiment, we invoked the `PbmQueryAssociatedProfile` API [3] on 2,048 VMs (having two disks each) that were associated with a storage policy.

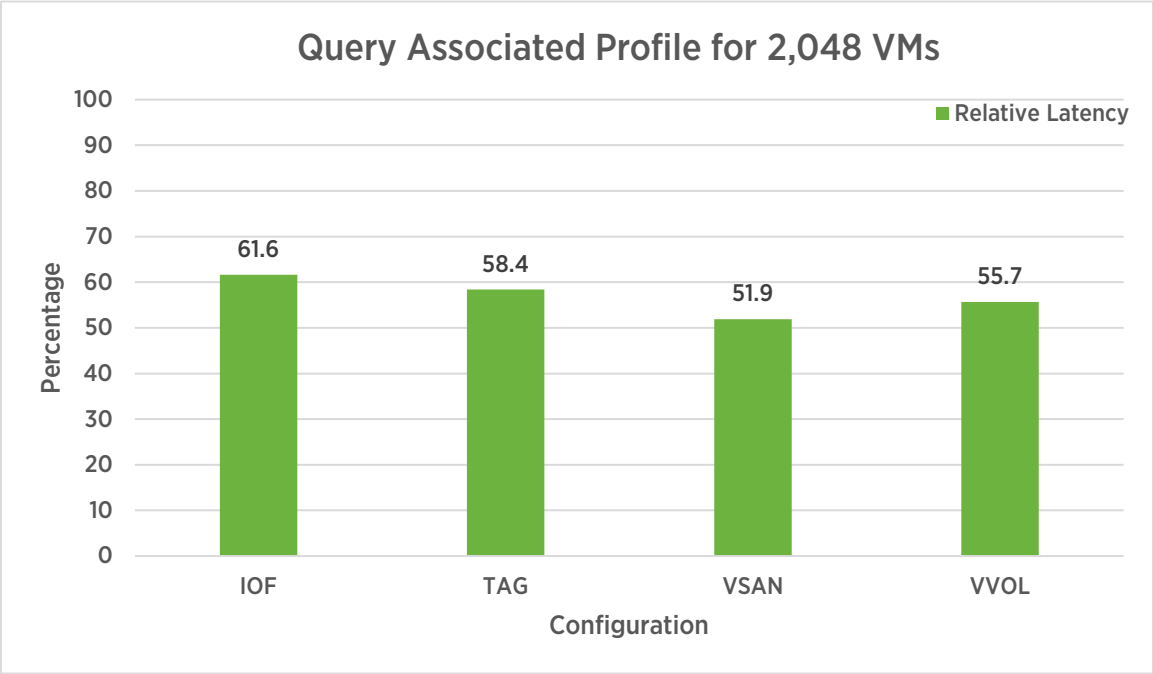

Figure 4. Latency of querying associated storage policy of a VM - vCenter Server 6.5 as a percentage of vCenter Server 6.0U3 (lower is better)

As we see in Figure 4, SPBM responds 38% to 48% faster when querying a policy associated with a VM/disk in vCenter Server 6.5 as compared to vCenter Server 6.0U3 for the same number of VMs per query.

Service Restart Time

Service restart time is the time taken by the SPBM service to become fully operational after it is restarted. A service restart occurs during a vCenter upgrade, during vCenter High Availability (VCHA) operations, or could be user initiated. Hence a lower service restart time would benefit all these scenarios.

We compared the SPBM service restart time in vCenter Server 6.5 against vCenter Server 6.0U3.

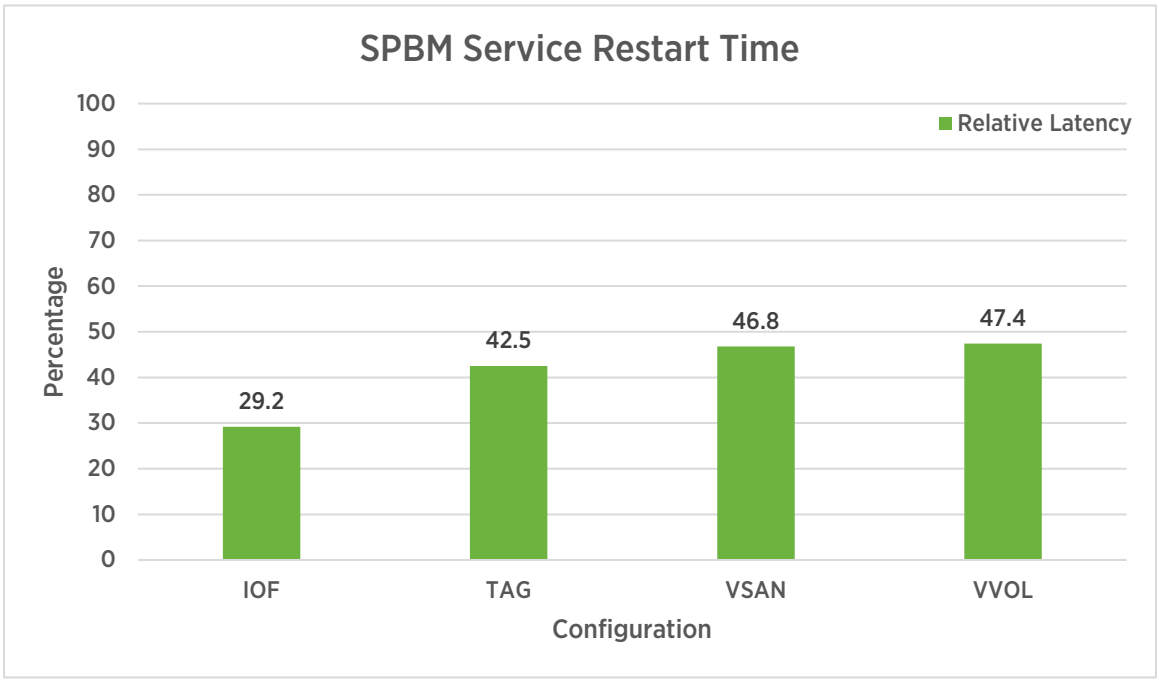

Figure 5. SPBM service restart time – vCenter Server 6.5 as a percentage of vCenter Server 6.0U3 (lower is better)

As shown in Figure 5, the SPBM service restarts in less than half the time in vCenter Server 6.5 when compared to 6.0U3.

Summary

There have been significant design changes and optimizations made to the SPBM service in vCenter Server 6.5. These changes help reduce the time taken by the SPBM service during critical vCenter Server operations. The performance optimizations are a combination of batching similar operations, introducing caching where appropriate, and optimizing the most common workflows. Since SPBM is in the critical path of frequently performed VM operations, these improvements will help provide a better user experience.

Scaling

This section considers the behavior of SPBM APIs with increasing SPBM inventory sizes like the number of storage policies and the number of policy associations in the system. The following experiments were run against vCenter Server 6.5.

Number of Storage Policies

In this section, we consider SPBM APIs that scale with the number of storage policies.

Query Profile

`API: PbmProfileProfileManager.PbmQueryProfile`

This API is used to get the storage policy IDs in vCenter Server. In this scaling experiment, we invoked this API with an increasing number of storage policies present in vCenter Server 6.5. The response times of this API was noted with batches of 64 storage policies all the way up to 1,024 policies in the system.

vSphere Client triggers:

1. Create/Clone/Edit/Relocate VM wizards
2. Policies & Profiles → VM Storage Policies

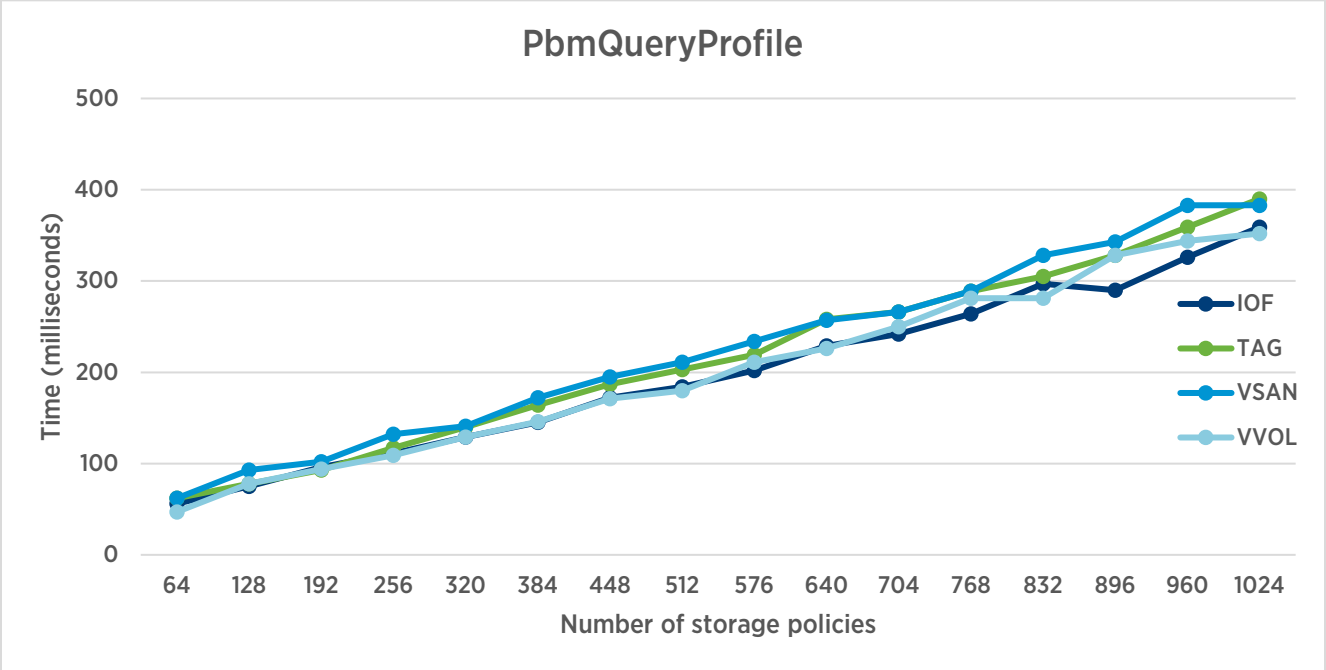

Figure 6. PbmQueryProfile times with increasing number of (user-defined) storage policies in vCenter Server 6.5

As we see in Figure 6, the time taken by the API increases linearly with the number of storage policies in vCenter Server. This trend indicates a healthy service with a predictable response time as the number of storage policies is scaled up.

Retrieve Content of a Storage Policy

`API: PbmProfileProfileManager.PbmRetrieveContent`

This API is used to fetch the complete contents of a given profile ID or multiple IDs. In this scaling experiment, we invoked this API with an increasing number of storage policies in the system.

vSphere Client triggers:

1. Create/Clone/Edit/Relocate VM wizard
2. Policies & Profiles → VM Storage Policies

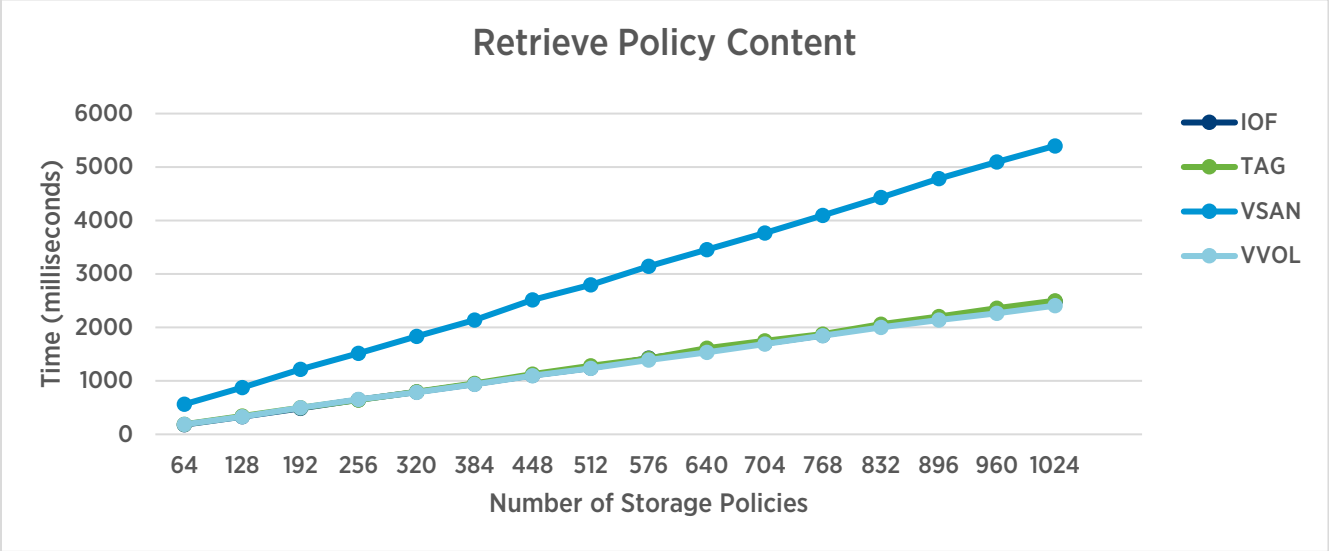

Figure 7. PbmRetrieveContent times with an increasing number of storage policies in vCenter Server 6.5

As we see in Figure 7, the time taken by the API increases linearly with the number of profile contents fetched using the API. vSAN scales more steeply than the other three configurations because the vSAN VASA provider publishes additional storage policies in the system.

Compliance Check of a Storage Policy

`API: PbmComplianceManager.PbmCheckCompliance`

This API triggers a fresh computation of compliance status of the storage objects with respect to its associated storage policy. This computation involves verifying whether the rules in the storage policy are being honored by the storage system. We did not observe any performance impact on compliance check with an increasing number of policies in vCenter Server 6.5.

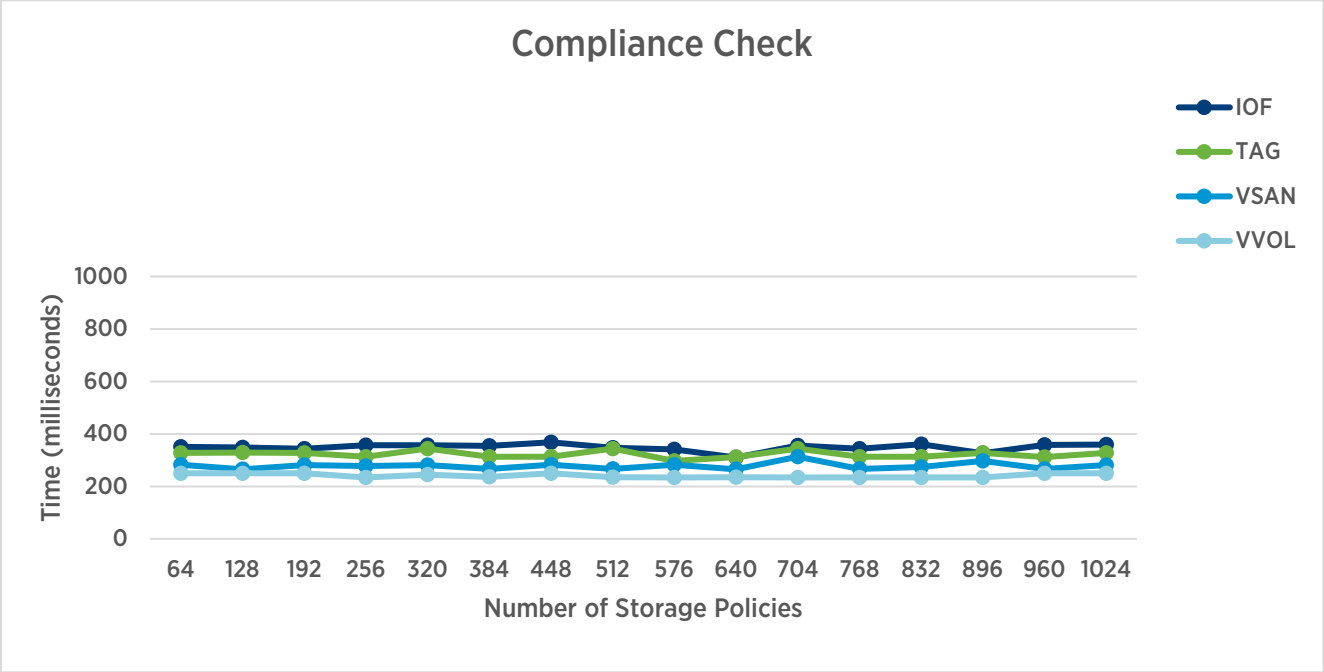

Figure 8. Compliance check times with increasing number of user-defined profiles in vCenter Server 6.5

Number of Associations

A storage policy can be associated with many VMs or disks. This section covers the APIs that are affected by an increase in such storage policy associations.

Compliance Operation

`API: PbmComplianceManager.PbmCheckCompliance`

This API triggers a fresh computation of compliance status for the storage objects. In this experiment, we passed an increasing number of VMs associated with the same storage policy to this API. We noted the latency in steps of 150 VMs.

vSphere Client triggers:

1. [Home] → Policies & Profiles → VM Storage Policies → [select storage policy] → Check Compliance
2. Select VM → Summary Page → VM Storage Policy → Check Compliance
3. Edit VM operation when changing the storage policy association

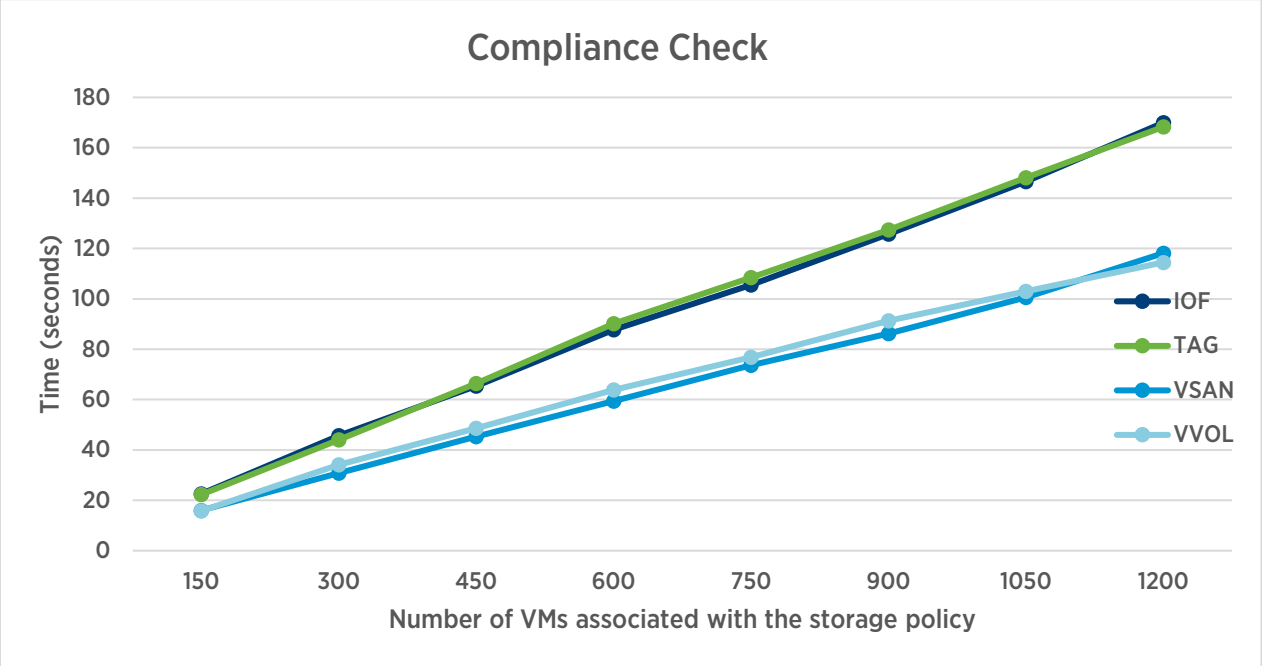

Figure 9. Compliance Check times with increasing number of VMs associated with the storage policy

As we see in Figure 9, the compliance check can take an increasing amount of time with the number of VMs associated with the given policy. The slope is linear, indicating a healthy response time with increasing scale. This also shows the performance optimizations we have made for vSAN and VVol.

Rollup Compliance Check

`API: PbmComplianceManager.PbmCheckRollupCompliance`

This API triggers a fresh computation of compliance status of a VM and its disks. It returns an aggregated result in the form of a rollup compliance status. In this experiment, we passed an increasing number of VMs associated with the same storage policy to this API. We noted the latency in steps of 150 VMs having 2 disks each.

UI triggers:

1. Create/Clone/Edit VM workflows
2. Policies & Profiles → VM Storage Policies

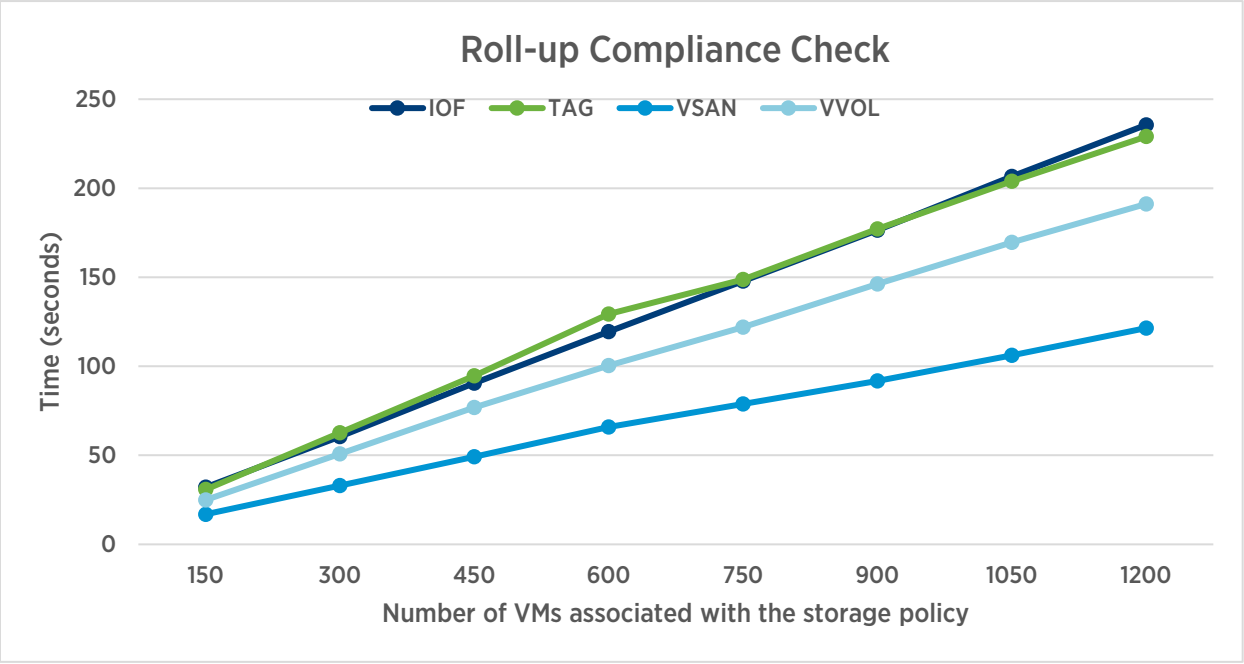

Figure 10. Roll-up Compliance Check times with increasing number of VMs associated with the storage policy

As we see in Figure 10, the time taken increases linearly with increasing number of storage objects associated with the storage policy.

Query Associated Profiles of Given Storage Entities

`API Name: PbmProfileProfileManager.PbmQueryAssociatedProfiles`

The PbmQueryAssociatedProfiles API returns storage policies associated with the given storage entities. For readings shown in Figure 11, we passed an increasing number of VMs associated with the same storage policy to PbmQueryAssociatedProfiles and noted the response times.

vSphere Client trigger:

[Any VM] → Configure → Policies

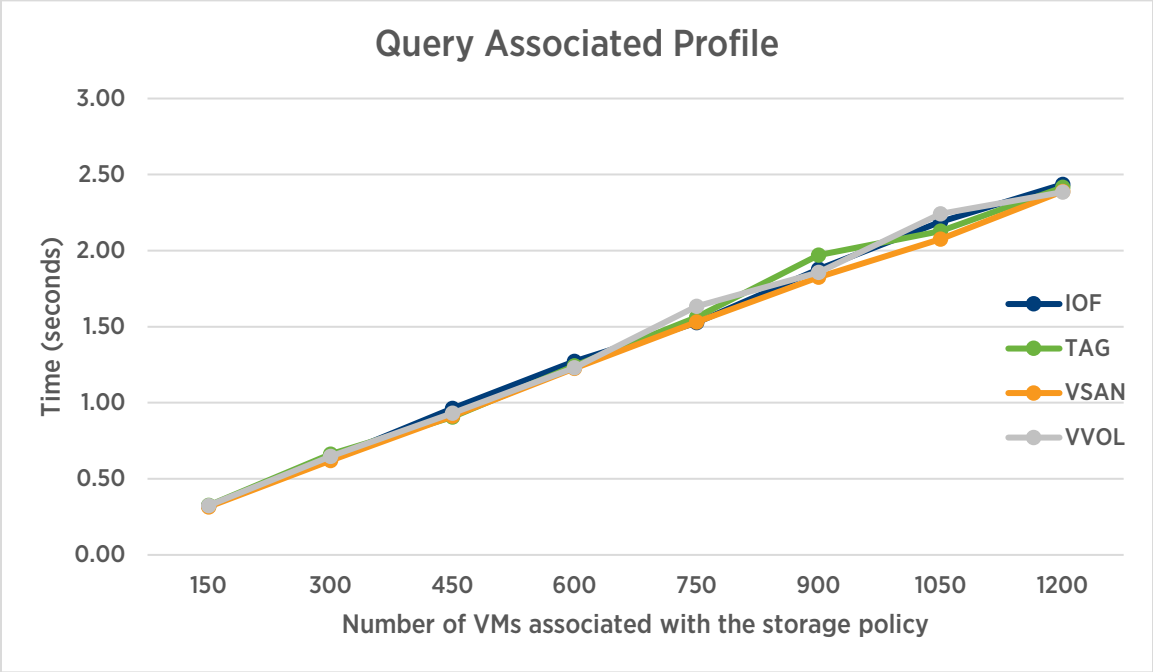

Figure 11. Query Associated Profile response time with increasing number of VMs

Figure 11 shows a healthy trend where PbmQueryAssociatedProfiles takes a linearly increasing amount of time with the number of VMs associated with the given policy.

Number of Datastores

This section details the APIs that are affected by an increase in datastores in the inventory.

Find Matching Datastores for a Storage Policy

`API: PbmPlacementSolver.PbmQueryMatchingHub`

This API is used to find matching datastores for the specified requirements profile. This method returns only those datastores that match the profile. In the following experiment, we increased the number of datastores as we noted the latency.

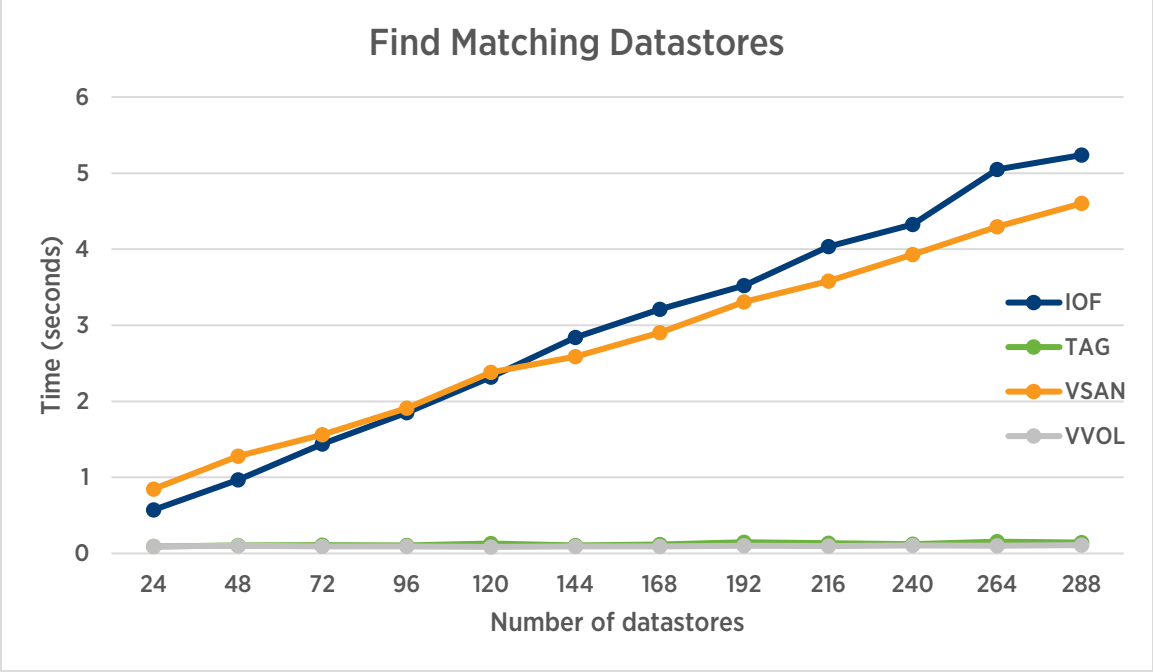

Figure 12. Latency of Find Matching Datastores for a given storage policy – with increasing number of datastores

As we see in Figure 12, the latency of finding datastores follows the expected linear trend with the number of datastores available in the inventory. It is noteworthy to specify that it is one of the best practices to pass along only the placement datastores of interest to be checked for compatibility.

Number of VASA Providers

SPBM Service Restart Time Scaling with Number of VASA Providers

SPBM service restart time depends on the number of VASA Providers that are registered with vCenter. As we noted earlier, service restart could occur during a vCenter Server upgrade or during vCenter High Availability (VCHA) operations. A faster restart time will benefit both of these scenarios. In this experiment, we restarted the service with a varying number of VASA providers and noted the time it took for the service to be ready again.

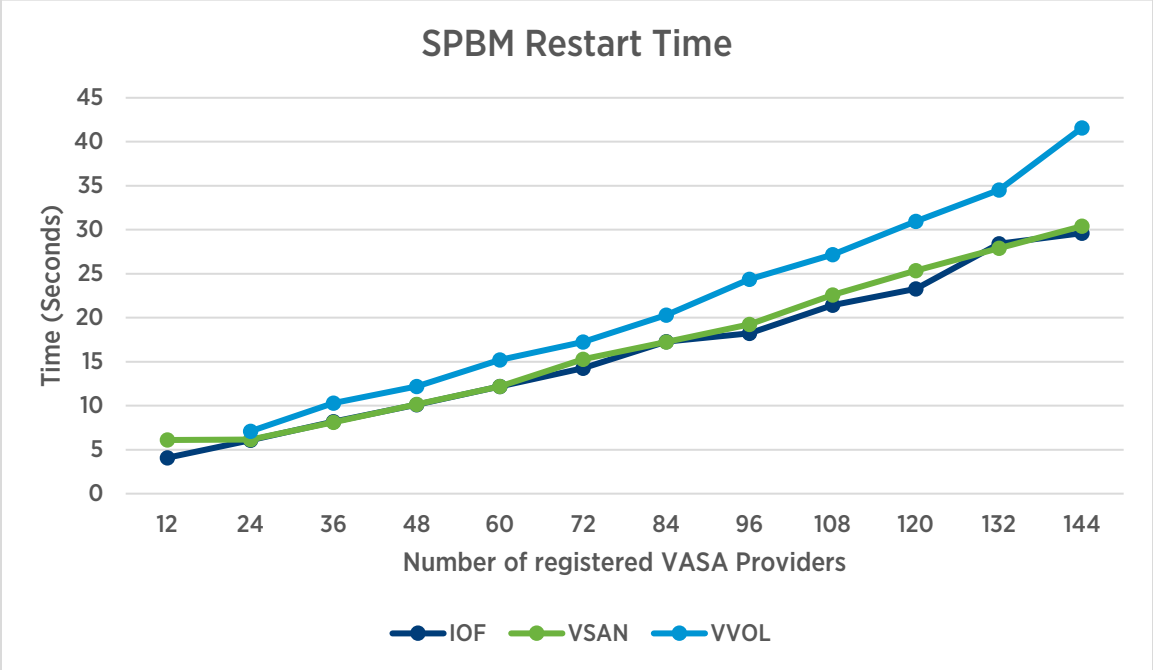

Figure 13. SPBM service restart time

As seen in Figure 13, the time it takes to restart the service follows a healthy linear trend with the number of registered VASA providers.

Conclusion

vCenter Server 6.5 comes with a much faster and more scalable SPBM service. We have optimized the most common operations in order to provide a better user experience.

Appendix

Testbed Setup

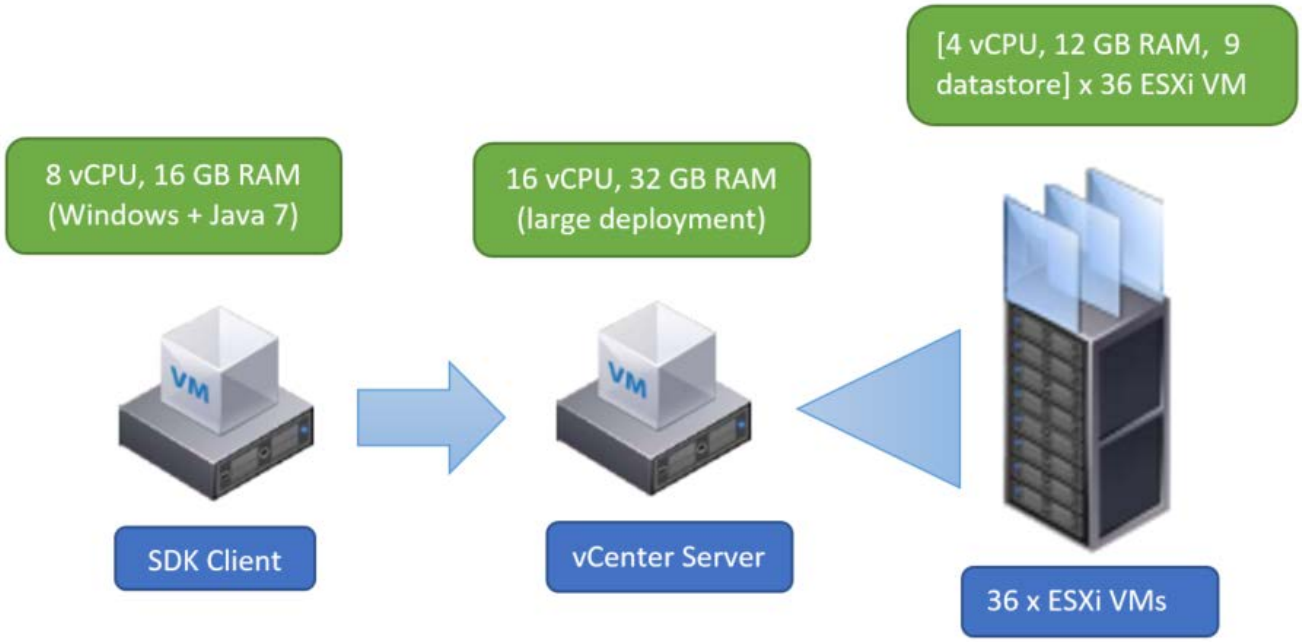

Figure 14. Testbed configuration

We sized our testbed within the configuration maximums for vSphere 6.5 [5].

Testbed Configuration

We set up our testbed with the following vSphere components:

VSPHERE COMPONENT	SOFTWARE VERSION	CONFIGURATION
vSphere Hosts	ESXi 6.0U3	VM: 4 vCPUs, 12GB vRAM
	ESXi 6.5	VM: 4 vCPUs, 12GB vRAM
vCenter Server Appliance	vCenter Server 6.0U3	VM: 16 vCPUs, 32GB vRAM
	vCenter Server 6.5	VM: 16 vCPUs, 32GB vRAM

Table 1. Testbed configuration

The servers hosting ESXi VMs, the vCenter server VM, and the SDK client VM are all of type Dell PowerEdge R730xd having Intel® Xeon® Processor E5-2680 v3 @ 2.50GHz with 256GB memory.

Benchmark Workload

The benchmark uses the vSphere Web Services SDK to perform SPBM-related operations. The operations are sequentially performed and latencies noted. After several iterations, the median of the noted values is chosen as the representative of the values. The experiments covered the most commonly used SPBM APIs. Table 2 lists the operations that we ran as part of our benchmark workload.

OPERATION	DESCRIPTION
Restart Service	Restart the SPBM service
Query Profiles	Query SPBM to return all existing policies' IDs
Query Content	Query SPBM to return all existing policies' content
Check Compliance	Check compliance of specified entities against specified policies
Check Rollup Compliance	For a specified VM, a rollup compliance check verifies the storage requirements of the VM and its virtual disks and compares them with the storage provider capabilities
Create VM with Policy	Create a VM with its disk, which is associated with a user-defined profile
Clone VM with Policy	Clone a VM with its disk, which is associated with a used-defined profile
Find Matching Hubs	Checks all the data stores in inventory to find out which ones are suitable for the given profile

Table 1. List of operations in our benchmark

Testbed Details

Common details - The following details are common for all configurations:

Rule-sets per Storage Policies	1
Rules per rule-set	4
Disks per VM	2
VMFS data stores per ESXi host	9
ESXi hosts per cluster	3
Additional vSAN datastores (as applicable)	12
Additional VVOL datastores (as applicable)	36

Table 3. Rules, disks, hosts, and so on that are common for all experiments

Setup used for comparative study – vCenter Server 6.5 compared against vCenter Server 6.0U3

Storage Policies	256
VMs	1,200
Additional vSAN data stores (as applicable)	144
Additional VVOL datastores (as applicable)	36

Table 4. Entities used during vCenter Server 6.5 vs. 6.0U3 comparison

Setup used for scaling experiment – vCenter Server 6.5

1. Scaling Storage Policies

Storage Policies	[0-1024] steps of 64
Clusters	12
VMs	0
VASA providers (TAG / IOF)	36
VASA providers (VSAN / VVol)	72

Table 5. Setup used for scaling experiments on vCenter Server 6.5

2. Scaling Associations

Storage Policies	256
Clusters	12
VMs	[0-1200] Steps of 150
VASA providers (TAG / IOF)	36
VASA providers (VSAN / VVol)	72

Table 6. Setup used for scaling experiments on vCenter Server 6.5

3. Scaling Datastores

Storage Policies	256
Clusters	[0-144] Steps of 12
VMs	0
VASA providers	[0-144] Steps of 12

Table 7. Setup used for scaling experiments on vCenter Server 6.5

4. Scaling VASA Providers

Storage Policies	0
Clusters	[0-144] Steps of 12
VMs	0
VASA providers	[0-144] Steps of 12

Table 8. Setup used for scaling experiments on vCenter Server 6.5

References

[1] VMware, Inc. (2013) VMware Storage Policy API 6.0.
https://code.vmware.com/apis/32/storage-policy

[2] VMware, Inc. (2017) vSphere APIs for I/O Filtering (VAIO) Program.
https://code.vmware.com/programs/vsphere-apis-for-io-filtering

[3] VMware, Inc. (2017) vSAN product page.
https://www.vmware.com/products/vsan.html

[4] VMware, Inc. (2017) Virtual Volumes product page.
https://www.vmware.com/in/products/vsphere/virutal-volumes.html

[5] VMware, Inc. (2017) Configuration Maximums for vSphere 6.5.
https://www.vmware.com/pdf/vsphere6/r65/vsphere-65-configuration-maximums.pdf

About the Authors

Shivashish Sahai is a performance engineer with the vCenter Server R&D group. He works on vCenter Server performance and scalability, with special focus on SPBM. Shivashish holds a bachelor's degree from the Dhirubhai Ambani Institute of Information & Communication Technology, India.

Adarsh Jagadeeshwaran works in the vCenter Server performance group in VMware, leading a team that is focused on vCenter Server resource management and scalability. Adarsh holds a master's degree in Computer Science from Syracuse University, Syracuse, USA.

Subramanian Neelakantan works as a developer in the SPBM team at VMware. He holds a master's degree in Computer Science from Sri Satya Sai Institute of Higher Learning, India. He has more than 17 years of experience working on cloud technologies at VMware and in previous jobs at IBM and Yahoo!.

Karthik Srinivas Sundararaman works as a developer in the SPBM team. He holds a bachelor's degree from B.S. Abdur Rahman Crescent Engineering College, Anna University, India. Karthik has more than 9 years of experience in the technology industry and has contributed to multiple VMware vCenter Server releases.

Acknowledgements

The authors would like to thank Ravi Soundararajan from the performance team for his guidance and feedback on this paper, Sandeep Sunny, Ganesh Patil, Atul Pandeyfor their review of the paper, and Julie Brodeur for her help in compiling the paper.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com