

VMWARE vSPHERE UPDATE MANAGER PERFORMANCE AND BEST PRACTICES

VMware vSphere 6.5

Table of Contents

- Introduction.....3**
- Experimental Configuration3**
 - Experimental Setup.....3
 - vSphere Update Manager and vCenter Server.....3
 - ESXi System.....3
 - Virtual Machine Operating Systems.....4
 - Experimental Configurations4
- Update Manager Server Deployment 5**
 - Best Practices5
- Performance for VUM Workflows 6**
 - Performance Tips7
- Remediation Concurrency7**
 - Performance Tips8
- Resource Consumption 8**
 - Performance Tips10
- Running VUM Operations with Workload.....10**
 - Performance Tips12
- Conclusion12**
- References12**

Introduction

VMware vSphere® Update Manager™ (also known as VUM) provides a patch management framework for VMware vSphere. IT administrators can use VUM to patch and upgrade VMware ESXi™ hosts, VMware Tools™, virtual hardware for virtual machines, as well as virtual appliances. As datacenters scale to accommodate more virtual resources, performance implications become more important for patch management. In this paper, we present a series of performance studies that cover various patch management workflows and their performance characteristics; we also provide best practice guidelines.

In the vSphere 6.5 release, VMware vSphere Update Manager has been integrated into the vCenter Server appliance (VCSA) for the Linux platform. The integration eliminates remote data transfers between VUM and VCSA while greatly simplifying the VUM deployment process. Thus, we also briefly present the performance and deployment differences between VUM for Windows and VCSA.

In summary, this paper covers the following topics:

- Experimental setup
- Various VUM deployment scenarios
- Operational performance for VUM workflows
- Resource consumption
- Running VUM tasks with workload

Experimental Configuration

Experimental Setup

vSphere Update Manager and vCenter Server

- Host Computer: Dell PowerEdge R730
- CPUs: 16 Intel Xeon Processors E5-2680 (2.5GHz)
- RAM: 32GB
- Hard drives: 512GB Local Hard Drive
- Network: E1000 Network Card (1Gbps)
- vSphere Update Manager: Version 6.5
- vCenter Server: Version 6.5

NOTE: VUM has been integrated in the vCenter Server 6.5 appliance for the Linux platform. The above configuration is used to run vCenter Server with VUM running as a service. VUM for Windows in the previous release was deployed in a dedicated server with 4 CPUs, 16GB memory, and 1G network bandwidth.

ESXi System

- Host Computer: Dell PowerEdge R730
- CPUs: Eight 2.5GHz Intel Xeon Processors E5-2680
- RAM: 32GB
- Hard drives: 100GB Local Hard Drive
- Network: E1000 Network Card (1Gbps)
- ESXi: Versions 6.5

Virtual Machine Operating Systems

- CentOS 6.8

Experimental Configurations

The configuration and environment used in the experiments are shown in Figure 1 (VUM on Windows for vCenter Server for Windows) and Figure 2 (VUM integrated as a service in vCenter Server 6.5 and later versions for Linux). If vCenter Server for Linux is used, there is no need to configure and deploy VUM. We use 32 ESXi servers to form a vMotion and DRS (fully automated) enabled cluster. On each host, we run 32 virtual machines (16 powered on) running CentOS 6.8.

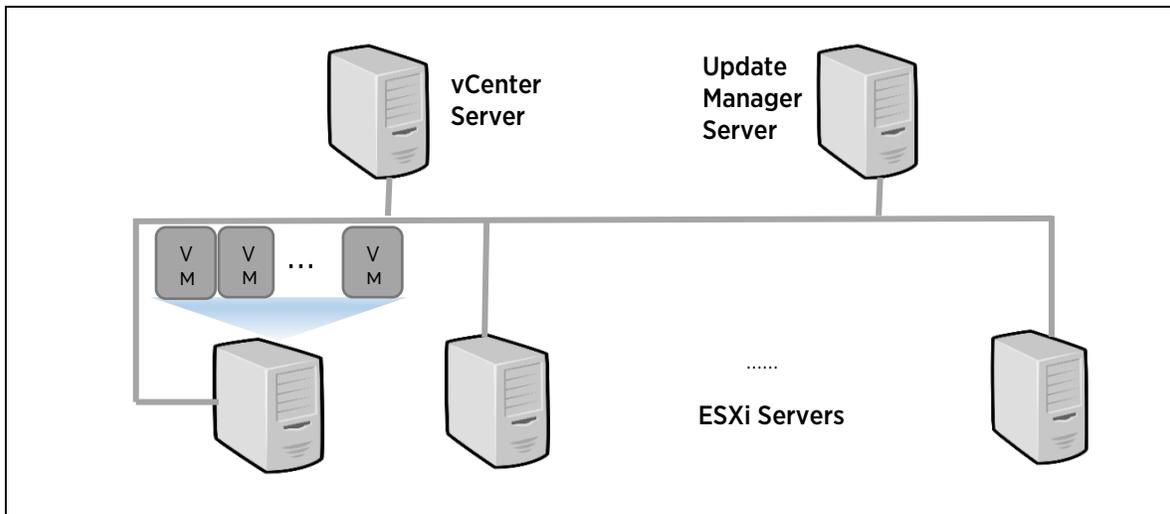


Figure 1. VUM on Windows for vCenter Server for Windows platform

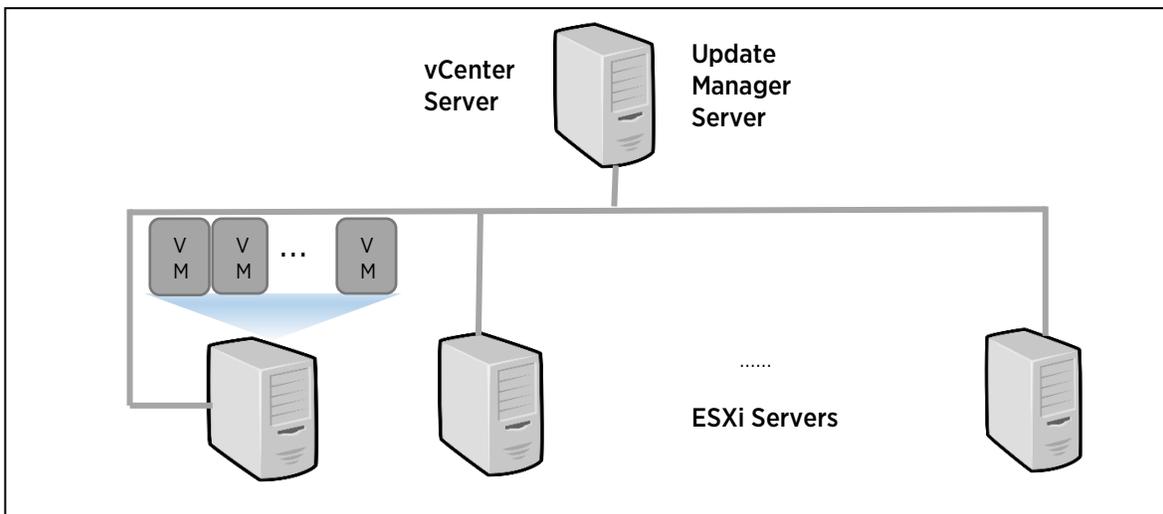


Figure 2. VUM integrated as a service in vCenter Server 6.5 for Linux platform

Update Manager Server Deployment

There are three Update Manager deployment models, as shown in Figure 3:

- Model 1 —In this model, vCenter Server and the Update Manager server share both a host and a database instance. This is the deployment model for VUM on vCenter Server 6.5 and later versions for the Linux platform.
- Model 2 —In this model, vCenter Server and the Update Manager server still share the same host, but use separate database instances.
- Model 3 —In this model, vCenter Server and the Update Manager server run on different hosts, each with its own database instance.

NOTE: Model 2 and model 3 are only supported by Update Manager on the Windows platform. On the Linux platform, Update Manager service and its database are integrated with vCenter Server in VCSA.

Although the Update Manager and vCenter Server run on the same machine in VCSA, the maximum supported inventory size and number of concurrent operations are the same as the Update Manager for Windows. In any deployment model, ensure that there are sufficient hardware sources for Update Manager and vCenter Server itself. Please consult the below best practices as a general guideline for resource requirement for Update Manager and subsequent sections in this paper for resource configuration recommendations with different inventory size. For resource configuration for vCenter Server itself, please refer to the *VMware vCenter Server Deployment Guide*.

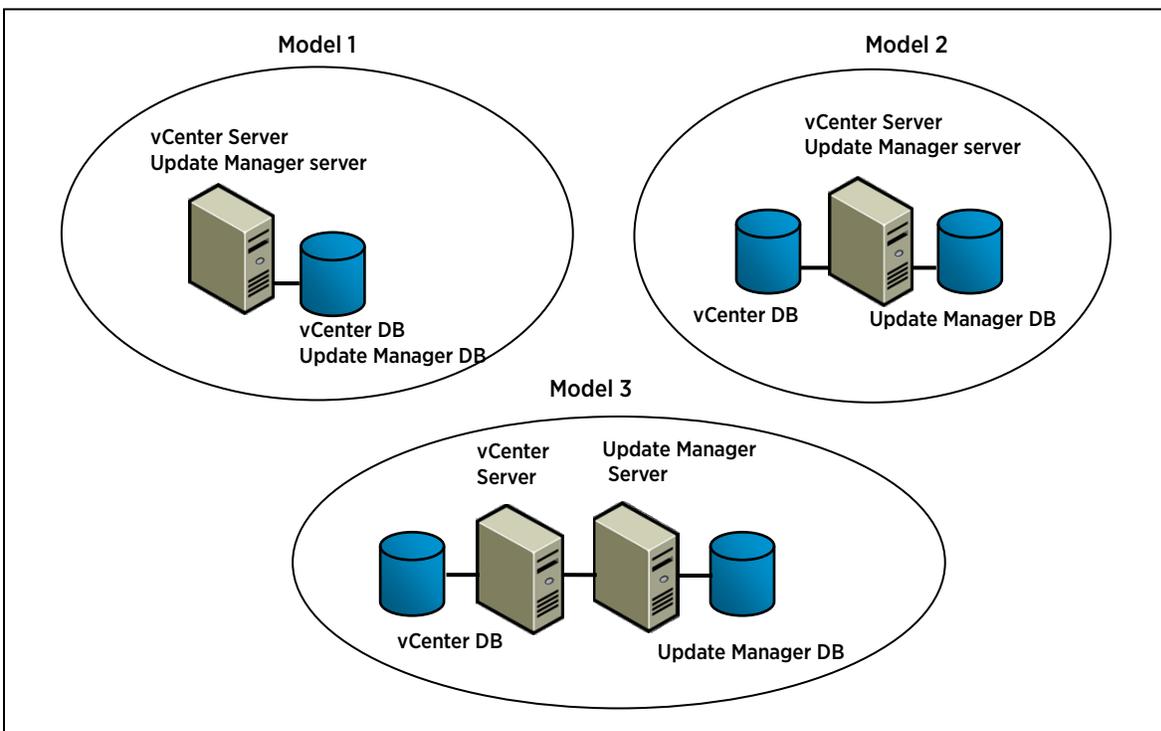


Figure 3. Host deployment models for Update Manager server

Best Practices

- For both ESXi and virtual machine patching, the Update Manager server transfers patch files over the network. To reduce I/O transactions, the Update Manager server caches patch files, some of which are several hundred of megabytes in size. Make sure the Update Manager service has at least 2GB of RAM to cache frequently used patch files in memory.

- Update Manager patch files can be large. Allocate separate physical disks for the Update Manager patch store if possible. In VCSA, the default location for Update Manager to store patch files is under `/storage/updatemgr/patch-store/` and a dedicated disk can be mounted to this path so that patch files do not affect the space that can be used by your other applications.
- For the integrated Update Manager with vCenter Server 6.5 on the Linux platform, make sure the database is given enough space for both the vCenter inventory and the Update Manager data. Initialization needs 150MB, and an upper estimate of monthly usage is around 90-100MB data in the database.

Performance for VUM Workflows

In this section, we present operational latency for single host and virtual machine update tasks. We also showcase the performance parity for Windows VUM with VCSA. For every operation we present here, we run the experiment in two different configurations: VUM on Windows running on a dedicated host (Figure 1), and VUM on Linux running on the same machine as vCenter Server for Linux (Figure 2). For configuration details please refer to the “Experimental Configuration” section. Table 1 shows the latency results for the following operations:

- ESXi host scan (scan a host for updates and patch information)
- Virtual machine scan (scan a virtual machine for update information)
- Stage host (download patches and update related metadata from the repository)
- Remediate host (update host patches)
- Remediate virtual machine (update tools and hardware version)

All numbers are averaged over multiple runs. Note that these latency numbers are only references based on experimental data on our configuration. The latency can be significantly affected by hardware configurations, network latency, and system workload.

	Scan Host	Scan VM	Stage the First Host	Stage another Host	Remediate Host	Remediate VM
VUM on Windows (Separated from vCenter)	6s	<1s	7m 24s	35s	5m 19s	41s
VUM on Linux (Collocated with vCenter)	3s	<1s	3m 38s	33s	5m 14s	36s

Table 1. Update Manager operation latency comparison between Windows and Linux (shown in minutes (m) and seconds (s))

As can be seen from Table 1, VUM on Windows vs Linux generally performs similarly for most of the operations. Note that staging the first host requires a larger amount of data to be downloaded, thus we separate the stage operation into two experiments: “Stage the First Host” and “Stage another Host.” For stage operations, VUM on Linux is considerably faster because placing VUM and vCenter Server in the same machine eliminates the remote file transfer incurred during the stage process. For the “Remediate Host” experiment in Table 1, our experiments show that powered-on virtual machines running on a host slow down the remediate process for the host by around 20% if we choose not to change the virtual machine’s power-on state. This is because the host needs to enter maintenance mode and reboot (automatically initiated by VUM) during the remediation, and thus the virtual machines that need to be online are migrated to other hosts, resulting in longer latency.

The results in Table 1 were measured on a low-latency local network setup. However, network latency plays an important role for most Update Manager operations. For the impact of a high-latency network, please refer to our prior study in the paper “VMware vCenter Update Manager 5.0 Performance and Best Practices.” For concurrent update performance, please refer to the “Remediate Concurrency” section.

Performance Tips

- Remediating a host is faster if there are no powered-on virtual machines. Powering off virtual machines and updating when the hosts are less loaded can help improve the remediation latency.
- Upgrading VMware Tools is faster if the virtual machine is powered on. Otherwise, Update Manager powers on the virtual machine before the VMware Tools upgrade. This might increase the overall latency.
- Upgrading virtual machine hardware is faster if the virtual machine is powered off. Otherwise, Update Manager powers off the virtual machine before upgrading the virtual hardware. This might increase the overall latency. **NOTE: Because VMware Tools must be up to date before virtual hardware is upgraded, Update Manager might need to upgrade VMware Tools before upgrading virtual hardware. In this case, the process is faster if the virtual machine is already powered on.**

Remediation Concurrency

In Update Manager 6.5, you can choose to remediate a cluster of hosts in sequence, in parallel, or in parallel with a given level of concurrency. Compared to sequential update, concurrent remediation that updates multiple hosts and virtual machines in parallel can reduce the total update time and service downtime. This section presents operational performance and several considerations for concurrent remediation.

An appropriate level of remediation concurrency can greatly shorten the update cycle for your entire vCenter Server inventory. However, performance benefits achieved by parallel remediation start to diminish beyond a certain limit due to resource (CPU and thread limit) constraints for VUM and vCenter Server. To demonstrate this effect, we use the setup described in the “Experimental Configuration” section and perform concurrent remediation by randomly picking different numbers of hosts and virtual machines to remediate.

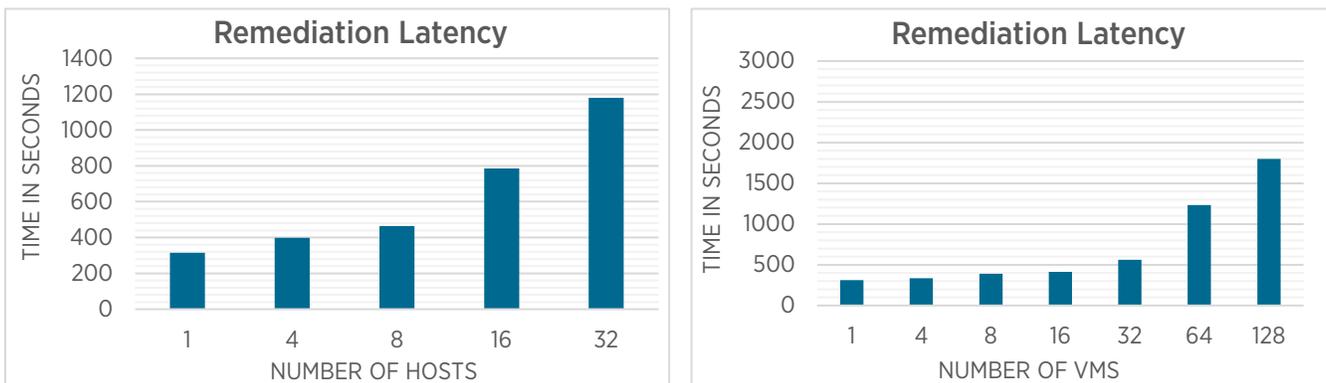


Figure 4. Concurrent remediate latency with different number of hosts and virtual machines (VMs)

Figure 4 shows the concurrent remediation latency for different numbers of hosts and virtual machines. As can be seen from the figure, under a certain concurrency limit (< 8 hosts and < 32 virtual machines), the latency scales very well. Upgrading hosts and virtual machines in parallel can greatly reduce the remediation latency in these cases. However, with too many hosts or virtual machines updated in parallel, latency starts to grow faster.

Due to the aforementioned performance reason and the maximum number of threads that can be used, Update Manager introduces Throttling Limits, which limit the maximum numbers of concurrent tasks. An update operation will be queued and performance will be affected if the concurrency of the operation exceeds its limit. Although those limits have been continuously scaling in newer releases to allow a larger number of parallel update operations, choosing the appropriate concurrency is critical for performance. Table 2 gives the recommended concurrency level for best performance. For the maximum limited concurrent operations in general, please consult the “Configuration Maximums” document for vSphere 6.5.

Update Manager Operations	Maximum Tasks per ESXi Host	Maximum Tasks per Update Manager Server
ESXi host scan	1	72
ESXi host remediation	1	48
VMware Tools scan	72	200
VM hardware scan	72	200
VMware Tools upgrade	30	128
VM hardware upgrade	30	128
ESXi upgrade	1	48

Table 2. Recommended concurrency for each Update Manager operation

Performance Tips

- Limiting the host remediation concurrency level to half of the number of hosts in the cluster can reduce vMotion intensity, often resulting in better overall host remediation performance. You can set this option by using the Update Manager remediation wizard.
- Do not remediate too many virtual machines at the same time. Significant update performance degradation is possible when virtual machine remediation concurrency reaches a certain level. As scan operations typically run much faster, you can scan more virtual machines concurrently in a cluster. Use Table 2 as a reference for how many hosts or virtual machines can be updated and upgraded simultaneously without hurting performance significantly.
- When all hosts in a cluster have all virtual machines powered off and are ready to enter maintenance mode, concurrent host remediation will typically be much faster than sequential host remediation as there is no live virtual machine migration overhead during the remediation process.

Resource Consumption

On the Linux platform, as Update Manager shares the same resources with the vCenter Server, it is important to ensure there are enough resources while performing update operations. In this section we present our study of VUM resource consumption in different scenarios.

Figure 5 shows the memory consumption and peak CPU utilization while several different update operations are being performed on Linux. VUM's memory usage is relatively stable and does not show a significant variation when different update operations are performed. As depicted in Figure 5, scan, stage, and remediate consumes 300 - 500MB of memory in VUM. Updating multiple objects concurrently does not severely increase the memory consumption in the short term. This is because VUM uses most of its memory to store the inventory information and cache patch files. The CPU utilization, however, can experience short-term spikes when performing highly concurrent tasks. For example, scanning and remediating 128 VMs triggered 300% - 400% CPU utilization, compared to only a small percentage when scanning and remediating one VM.

VMWARE vSPHERE UPDATE MANAGER PERFORMANCE AND BEST PRACTICES

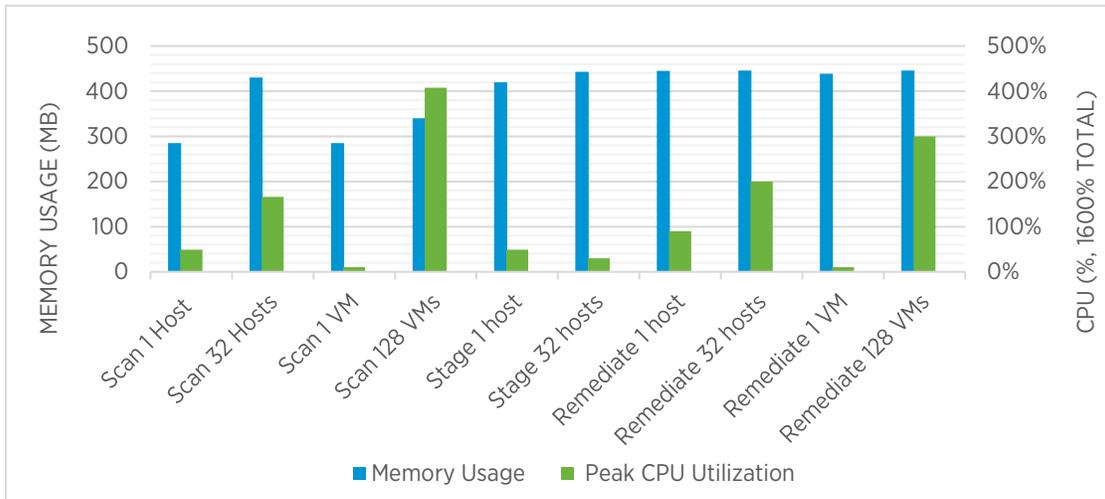


Figure 5. Memory and CPU consumption for Update Manager on Linux for update operations

VUM resource usage is affected by inventory size. Figure 6 shows VUM’s memory and vCenter Server’s database space consumption with different inventory sizes after all the hosts and VMs are updated. As can be seen in the figure, although the memory and database space consumption do not grow linearly, they are significantly affected by the inventory size. The trend shown in Figure 6 provides a way to estimate the resource consumption for a given inventory size. Ensure that enough memory and database space are provided for your inventory on VCSA.

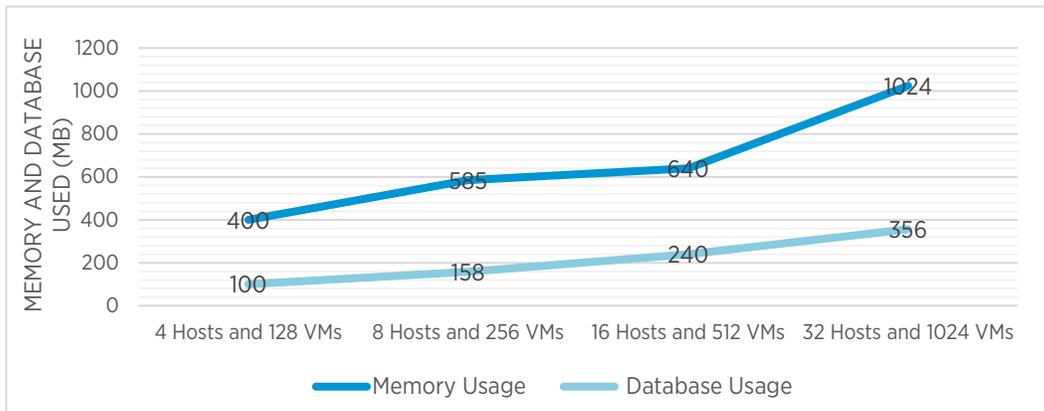


Figure 6. Memory and database space consumption for Update Manager on Linux with different inventory

Based on the resource usage trend we studied and the *VMware vCenter Server Deployment Guide*, we provide Table 3 as a resource configuration guide for different inventory sizes for VCSA. Please refer to the *VMware vCenter Server Deployment Guide* for general deployment and resource requirements for your environment.

	SMALL: UP TO 100 HOSTS/ 1,000 VIRTUAL MACHINES	MEDIUM: UP TO 400 HOSTS/ 4,000 VIRTUAL MACHINES	LARGE: UP TO 1,000 HOSTS/ 10,000 VIRTUAL MACHINES
CPU	4	8	16
MEMORY	16 GB (~1G for VUM)	24 GB (~2G for VUM)	32 GB (~3G for VUM)

Table 3. Resource requirements for VCSA with VUM

Performance Tips

- The amount of memory consumed by Update Manager is primarily affect by the size of the inventory, in particular the number of virtual machines. Based on the memory usage projection, make sure Update Manager can consume the corresponding amount of memory resources in your system.
- During update/upgrade, memory usage grows modestly with more concurrent update/upgrade operations, and CPU usage can experience short-term spikes.
- Monitor the resource usage by VUM as well as other services periodically using *top*, *htop*, or other tools, especially when the system is under heavy workload. Update the resource or configuration when necessary.

Running VUM Operations with Workload

To understand how workloads (for example, VM provisioning operations) in the system and VUM operations can affect each other, we deploy two clusters, each of which contains 8 hosts and 256 VMs (128 powered on) in the inventory. We compare various update and upgrade performance when the system is idle to the case where there are VM provisioning workloads. The workloads are injected into one cluster by a benchmark with 4 clients, which are injected into one cluster by a benchmark with 4 clients, which perform VM-related tasks including

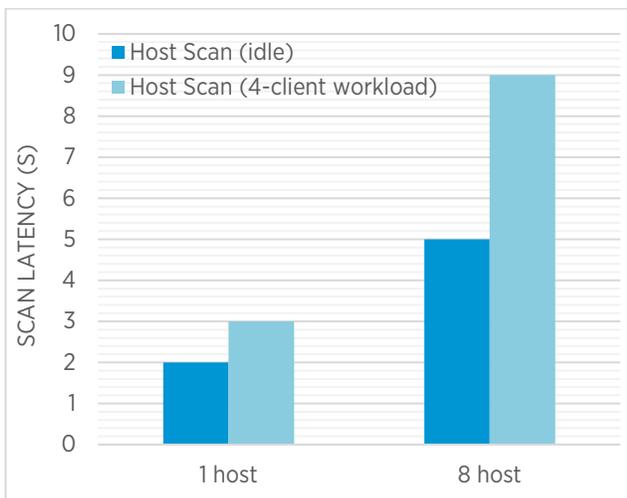


Figure 7. Host scan latency with different workloads

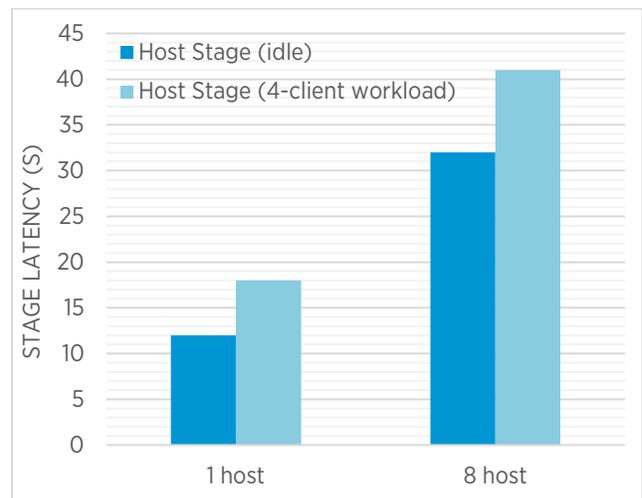


Figure 8. Host stage latency with different workloads

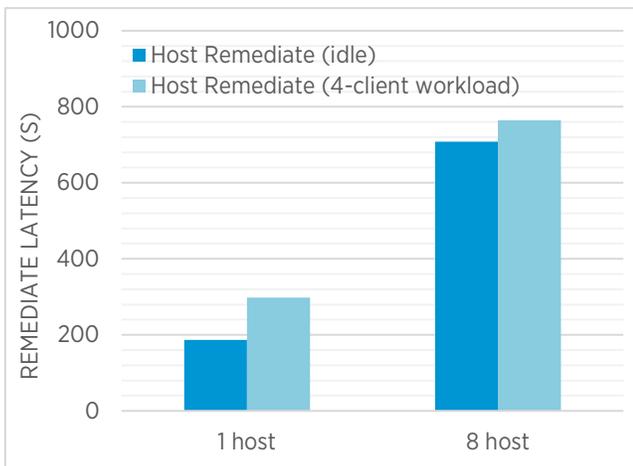


Figure 9. Host remediate latency with different workloads

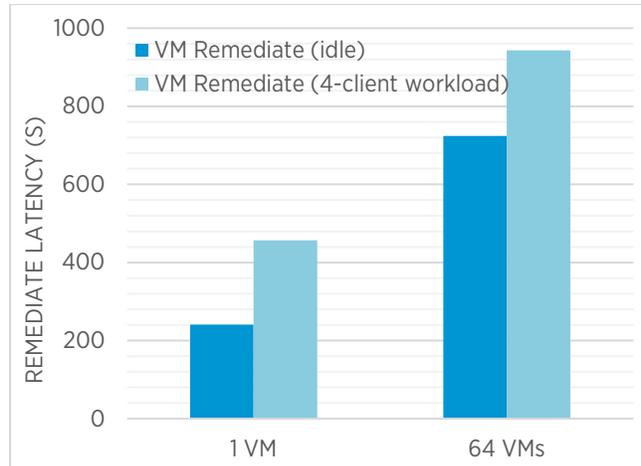


Figure 10. VM remediate latency with different workloads

powering on, VM migration, VM cloning, reconfiguration, and so on, in parallel. On another cluster, we perform VUM update operations.

Figures 7, 8, and 9 show the performance for scan, stage, and remediate under different workloads. As can be seen from the charts, when the system is idle, VUM operations are considerably faster. This is true no matter whether you update one host or multiple hosts in parallel. Depending on the workload currently handled by vCenter Server, host-related update/upgrade performance can be slowed down by around 10% to 40%. Remediating VMs exhibits similar latency increase, as shown in Figure 10. On the other hand, updating VMs and hosts can affect your running workloads, as reboot is required during the remediation process. Additionally, updating operations needs vCenter Server's involvement to perform virtual machine migration, power on, relocate, and so on, which can slow down other ongoing vCenter Server tasks.

To quantify VUM operations' impact on vCenter Server provisioning tasks, Figure 11 shows the latency for the benchmark's provisioning operations under VUM workload. In particular, powering on and off VMs are slowed down by more than 20% due to the added VUM workload. Registering VMs and snapshot operations are also considerably affected. Most of the other VM provisioning operations are not affected significantly by VUM operations.

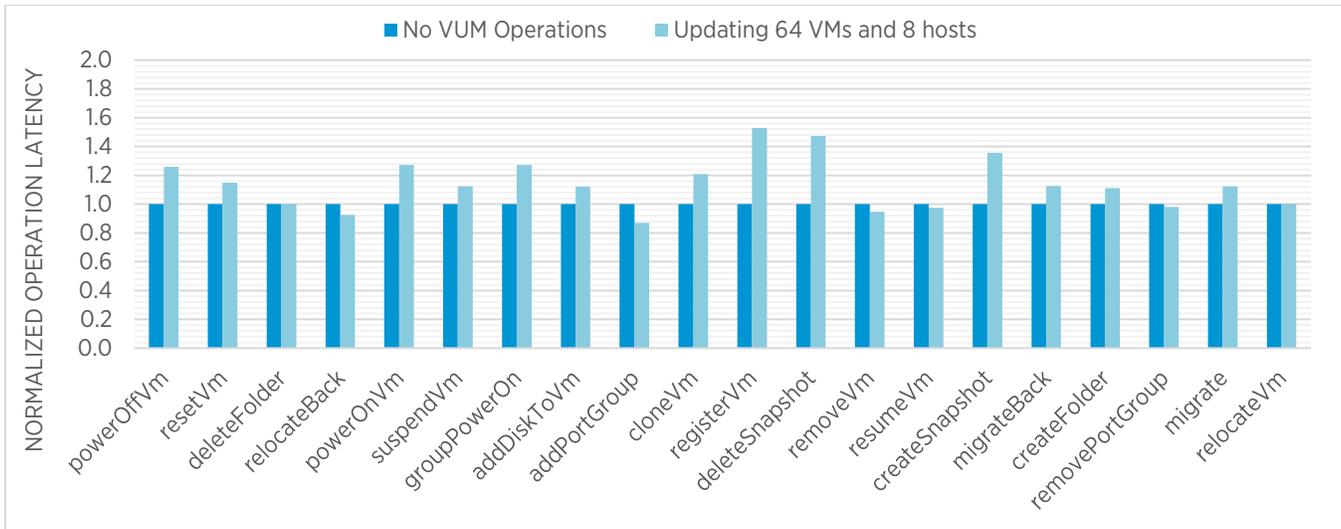


Figure 11. VUM's impact on VM operation latency

Figure 12 summarizes the overall benchmark throughput. When the system does not run any VUM tasks, the throughput of the benchmark is 27.9 operations per minute on average. By contrast, the average benchmark throughput is 27.3 when VUM handles update tasks for 64 virtual machines and 8 hosts, a 2.2% degradation compared to the case where VUM is idle. Based on the results shown in Figures 11 and 12, we can conclude that if VUM operations and VM provisioning tasks run on two different clusters, the performance of VUM and vCenter tasks do not affect each other significantly. In another experiment where we run VUM and benchmark operations against the same clusters, the throughput drops by 10%. Therefore, it is recommended that VUM operations are performed on a cluster that is not handling other vCenter Server tasks at the same time.

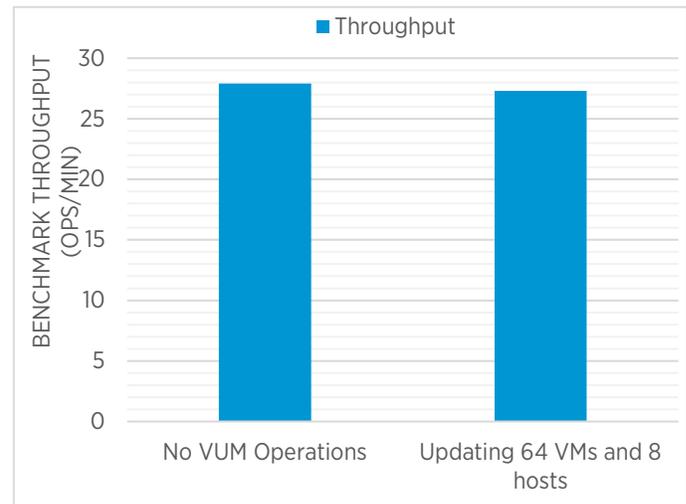


Figure 12. VUM's impact on throughput

Performance Tips

- It is highly recommended that update and upgrade operations are performed while the system is under low workload to ensure minimum impact on the online services. Cluster remediation is most likely to succeed when the cluster is utilized at no more than 80%. For heavily-used clusters, cluster remediation is recommended to be performed during off-peak periods, when workload is not heavy.
- When possible, it is best to suspend or power off some virtual machines before the operation starts. If you have more than one cluster, direct the requests or workloads to other clusters before upgrading the cluster.

Conclusion

VMware vSphere Update Manager delivers the most full-featured and robust patch management product for vSphere 6.5. This white paper presents experimental results and recommends various performance tips to help your Update Manager operations run as efficiently as possible.

References

1. *Installing and Administering VMware vSphere Update Manager*. VMware, Inc., 2011.
http://pubs.vmware.com/vsphere-50/topic/com.vmware.powercli.vum_inst.doc_50/vumps_admgd_preface.html.
2. *VMware vSphere Update Manager Database Sizing Estimator*. VMware, Inc., 2011.
<http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-update-manager-50-sizing-estimator.xls>.
3. "Configuration Maximums vSphere 6.0." VMware, Inc., 2016.
<https://www.vmware.com/pdf/vsphere6/r60/vsphere-60-configuration-maximums.pdf>
4. "VMware vCenter Update Manager 5.0 Performance and Best Practices." VMware Inc., 2011.
<http://www.vmware.com/techpapers/2011/vmware-vcenter-update-manager-50-performance-and-10208.html>

About the Authors

Yong Li is a senior member of technical staff in VMware and achieved his PhD in Computer Engineering from the University of Pittsburgh in 2013. Since he joined VMware, he has worked on several vCenter Server related projects, some of which include the Update Manager performance study, enhancing scalability and performance of vCenter Server, and vCenter Server in linked mode.

Acknowledgements

We would like to thank the Update Manager team for their tremendous support. Without their help, this white paper would not have been possible. Additionally, thanks are given to Lei Chai, Priya Sethuraman, Boris Valkov, Xuwen Yu, and Ravi Soundararajan for their valuable comments and feedback. Finally, we would like to thank Julie Brodeur for her help on editing and improving this white paper.



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

Copyright © 2016 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Comments on this document: <https://communities.vmware.com/docs/DOC-32941>