PERFORMANCE STUDY – NOVEMBER 2016

DRS PERFORMANCE

VMware vSphere 6.5

vmware®

Table of Contents

Introduction

VMware vSphere® Distributed Resource Scheduler™ (DRS) is more than a decade old and is constantly innovating with every new version. In vSphere 6.5, DRS comes with many new features and performance improvements to ensure more efficient load balancing and VM placement, faster response times, and simplified cluster management.

In this paper, we cover some of the key features and performance improvements to highlight the more efficient, faster, and lighter DRS in vSphere 6.5.

New Features

Predictive DRS

Historically, vSphere DRS has been reactive—it reacts to any changes in VM workloads and migrates the VMs to distribute load across different hosts. In vSphere 6.5, with VMware vCenter Server® working together with VMware vRealize® Operations™ (vROps), DRS can act upon predicted future changes in workloads. This helps DRS migrate VMs proactively and makes room in the cluster to accommodate future workload demand.

For example, if your VMs' workload is going to spike at 9am every day, predictive DRS will be able to detect this pattern before-hand based on historical data from vROPs, and can prepare the cluster resources by using either of the following techniques:

- Migrating the VMs to different hosts to accommodate the future workload and avoid host over-commitment
- Bringing back a new host from stand-by mode using VMware vSphere® Distributed Power Management™ (DPM) to accommodate the future demand

How It Works

To enable predictive DRS, you need to link vCenter Server to a vROps instance (that supports predictive DRS), which monitors the resource usage pattern of VMs and generates predictions. Once vROps starts monitoring VM workloads, it generates predictions after a specified learning period. The generated predictions are then provided to vCenter Server for DRS to consume.

Once the VMs' workload predictions are available, DRS evaluates the demand of a VM based on its current resource usage and predicted future resource usage.

Demand of a VM = Max (current usage, predicted future usage)

Considering the maximum of current and future resource usage ensures that DRS does not clip any VM's current demand in favor of its future demand. For the VMs which do not have predictions, DRS computes resource demand based on only the current resource usage.

Look Ahead Interval

The predictions that DRS gets from vROps are always for a certain period of time, starting from the current time. This period is known as the "look ahead interval" for predictive DRS. This is by default 60 minutes starting from the current time, which means, by default the predictions will always be for the next one hour. So if there is any sudden spike that is going to happen in the next one hour, predictive DRS will detect it and will prepare the cluster to handle it.

Case Study

The following scenario shows how DRS, based on predictions, migrates VMs proactively to avoid future load imbalance. We used a cluster of four hosts, where one of them (host 10.156.234.43) ran a VM whose workload spiked occasionally. Then we linked the vCenter to a vROps instance to monitor and generate predictions after the learning period. As you can see in Figure 1, vROps generated predictions for the VM workload (which is highlighted by the red circle)

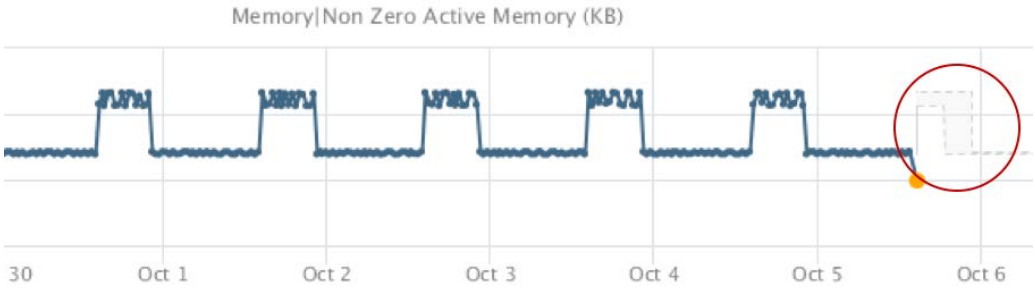


Figure 1 - Virtual Machine CPU usage pattern

After vROps pushed the generated predictions into vCenter Server, even though the cluster load is fairly distributed, you can see that DRS migrated VMs from host 10.156.234.43 to accommodate the predicted future workload spike.

Figure 2 shows the load distribution across hosts before the migrations.

Name	1 ▲	Cluster	Consumed CPU %	Consumed Memory %
10.156.234.43		BLR-Cluster	38	18
10.156.234.7		BLR-Cluster	39	18
10.156.237.129		BLR-Cluster	39	18
10.156.237.232		BLR-Cluster	38	16

Figure 2 - Host load distribution

Figure 3 shows the list of migrations that happened.

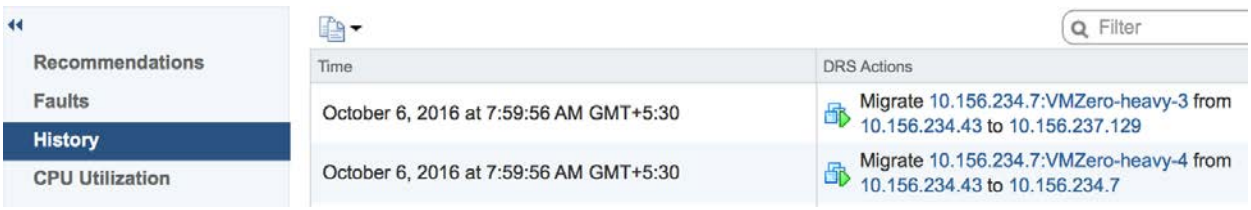


Figure 3 - DRS initiates migrations based on predictions from vROps

Figure 4 shows decreased load on host 10.156.234.43 due to VM migrations.

Name	1 ▲	Cluster	Consumed CPU %	Consumed Memory %
10.156.234.43		BLR-Cluster	25	11
10.156.234.7		BLR-Cluster	47	18
10.156.237.129		BLR-Cluster	47	18
10.156.237.232		BLR-Cluster	39	15

Figure 4 - Host Load distribution after migration

This case study shows that predictive DRS migrates VMs proactively to accommodate future workload spikes.

Network-Aware DRS

Traditionally, DRS has always considered the compute resource (CPU and memory) utilizations of hosts and VMs for balancing load across hosts and placing VMs during power-on. This generally works well because in many cases, CPU and memory are the most important resources needed for good application performance. However, since network availability is not considered in this approach, sometimes this results in placing or migrating a VM to a host which is already network saturated. This might have some performance impact on the application if it happens to be network sensitive.

DRS is network-aware in vSphere 6.5, so it now considers the network utilization of host and network usage requirements of VMs during initial placement and load balancing. This makes DRS load balancing and initial placement of VMs more effective.

How It Works

During initial placement and load balancing, DRS first comes up with the list of best possible hosts to run a VM based on compute resources and then uses some heuristics to decide the final host based on VM and host network utilizations. This makes sure the VM gets the network resources it needs along with the compute resources.

The goal of network-aware DRS in vSphere 6.5 is only to make sure the host has sufficient network resources available along with compute resources required by the VM. So, unlike regular DRS, which balances the CPU and memory load, network-aware DRS **does not** balance the network load in the cluster, which means it will not trigger a vMotion when there is network load imbalance.

Network-Aware Initial Placement

DRS does initial placement in two steps:

1. It compiles the list of possible hosts based on cluster constraints and compute resource availability and ranks them.
2. Then, from the list of hosts, it picks the host with the best rank and best network resource availability.

Network-Aware Load balancing

During a load balancing run, DRS

1. First generates the list of possible migration proposals.
2. Then eliminates the proposals whose destination hosts are network saturated.
3. From the remaining list of proposals, recommends the one with the maximum balance improvement in terms of compute resources and that also contributes to network resource availability on the source host, in case the source host is network saturated.

Host Network Saturation Threshold

As mentioned earlier, DRS will avoid a network loaded host during load balancing decisions, only if its network utilization is beyond a certain threshold. This threshold is set to 80% by default. So, unless the host network utilization is above 80%, DRS considers the host to be a good candidate in terms of network resource availability.

If a host's network utilization is at or above the saturation threshold, DRS considers it to be network saturated. If all the hosts in the cluster are network saturated, DRS will prefer **not** to migrate VMs with network load, since migrating network loaded VMs to an already network saturated host would result in further degradation of VM performance. When DRS cannot migrate VMs due to this behavior, this can sometimes result in an imbalanced cluster.

Monitoring Host Network Utilization

Starting in vSphere 6.5, you can monitor the host network load distribution under the DRS monitoring tab in the vSphere Web Client as shown in Figure 5.

The network utilization percentage of a host is the average capacity that is being utilized across all the physical NICs (pNICs) on that host. For example, if a host has three pNICs, one of them is 90% utilized and the other two are 0% utilized, then the network utilization of the host is considered to be 30%.

Figure 5 - Network load distribution view in the vSphere Web Client

Avoiding Network-Saturated Host During Initial Placement

The following case study shows how network-aware DRS avoids a host with high network utilization during initial placement of VMs.

We started with a cluster of four hosts having very similar resource utilization. Figure 6 and Figure 7 show CPU and memory utilization across the four hosts.

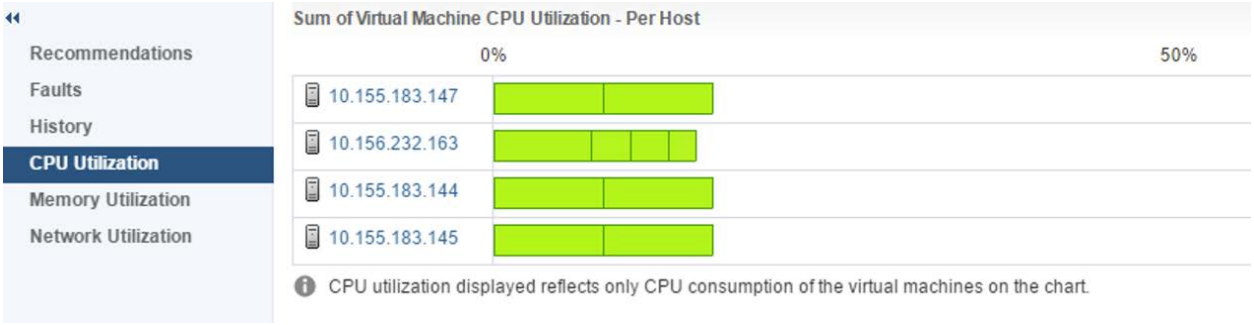


Figure 6 - CPU utilization view shows even distribution

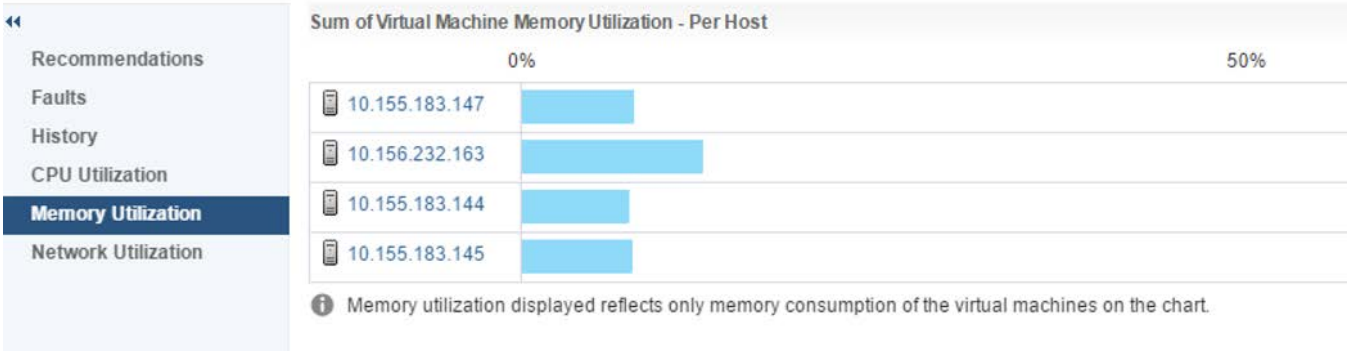


Figure 7 - Memory utilization view shows mostly even distribution

As you can see in Figure 8, one of the hosts (10.152.232.163) has high network utilization.

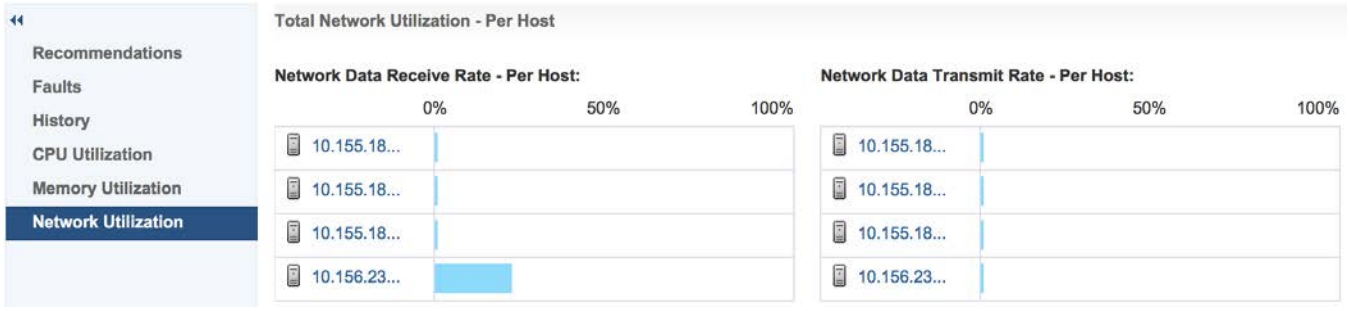


Figure 8 - Network Utilization shows one of the hosts is having high network utilization

At this point, we introduced (powered-on) eight new VMs into the cluster and noticed that DRS avoided the host with high network utilization while placing the VMs. Figure 9 shows the increase in CPU utilization of the other 3 hosts due to these newly powered-on VMs, while the CPU utilization in the network saturated host remains the same.

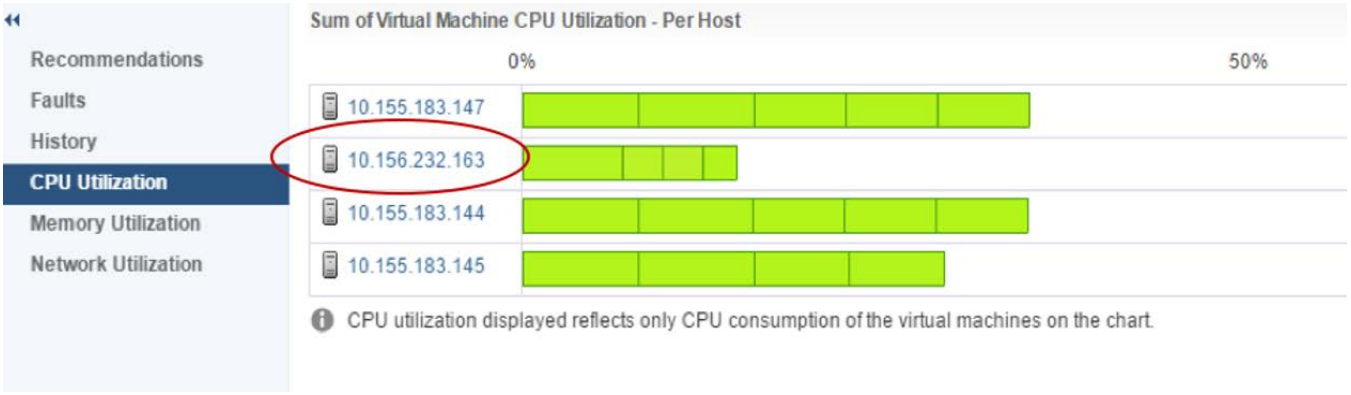


Figure 9 - CPU utilization view shows the distribution after introducing 8 new VMs

Avoiding Network Saturated Host During Load Balancing

The following case study explains how network-aware DRS avoids a host with higher network utilization while balancing the cluster load.

For this study, we started with a cluster of four hosts. One of the hosts is network-saturated, with a utilization of over 80%. Figure 10 shows the network utilization view in the cluster.

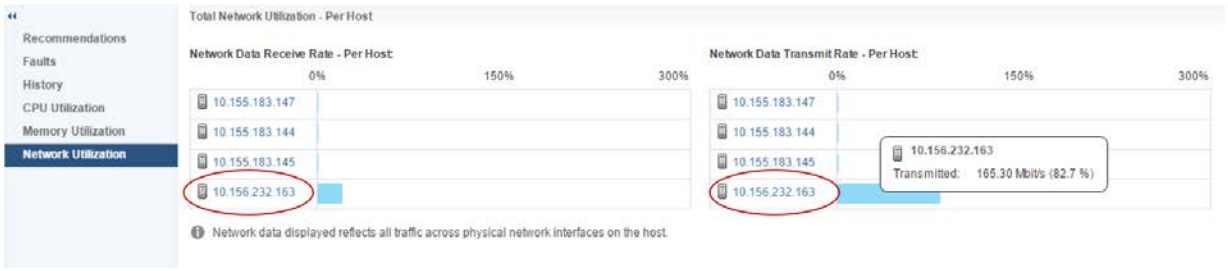


Figure 10 - Network utilization view with one host being network saturated

As we can see in Figure 11, the cluster is imbalanced. There is an uneven distribution of CPU load in the cluster, and the network saturated host is utilizing the least CPU. Figure 12 shows the CPU load distribution across the hosts.

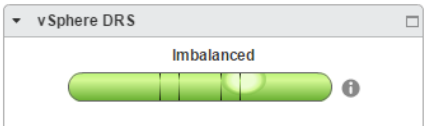


Figure 11 - Cluster Balance view

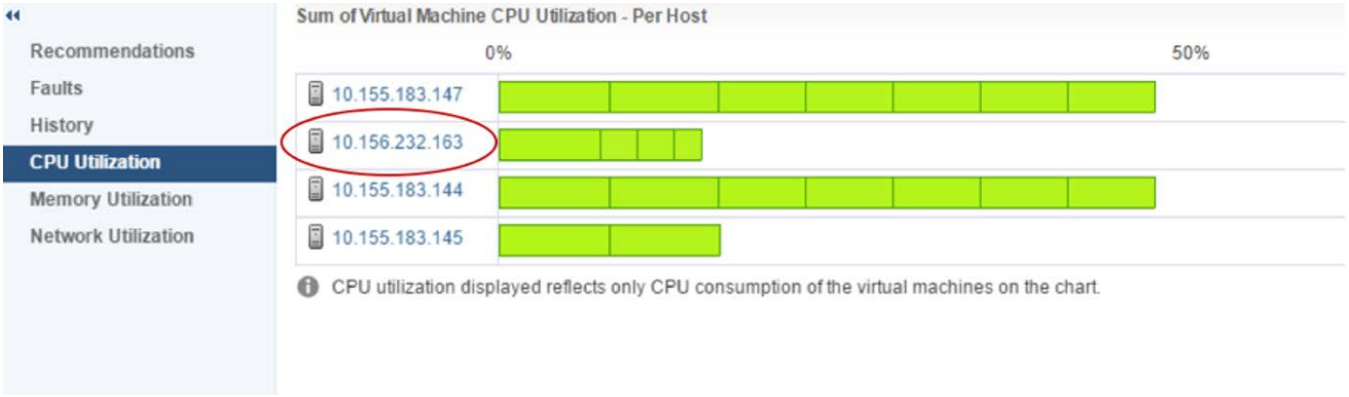


Figure 12 - CPU utilization view showing uneven load distribution

In order to balance the CPU utilization across the cluster, DRS initiates VM migrations. Figure 13 shows the list of generated migrations where DRS completely avoids the network saturated host despite it being the least utilized in terms of CPU.

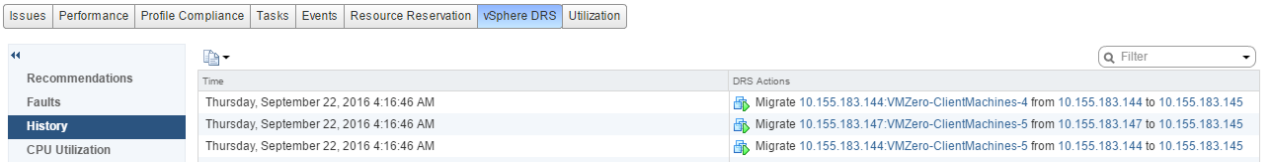


Figure 13 - Recommendations history view showing load balancing recommendations when avoiding network saturated hosts

The CPU utilization remains unchanged in the network saturated host, since DRS avoided moving VMs into the host (Figure 14).

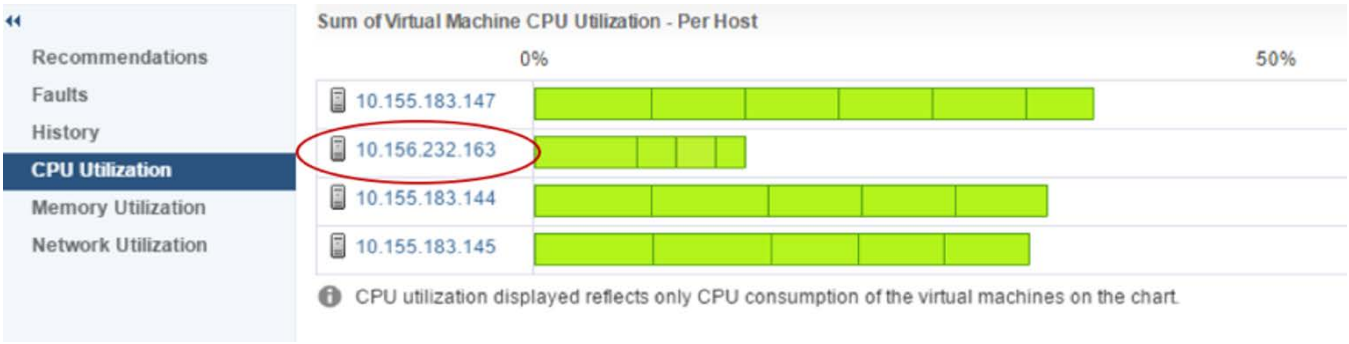


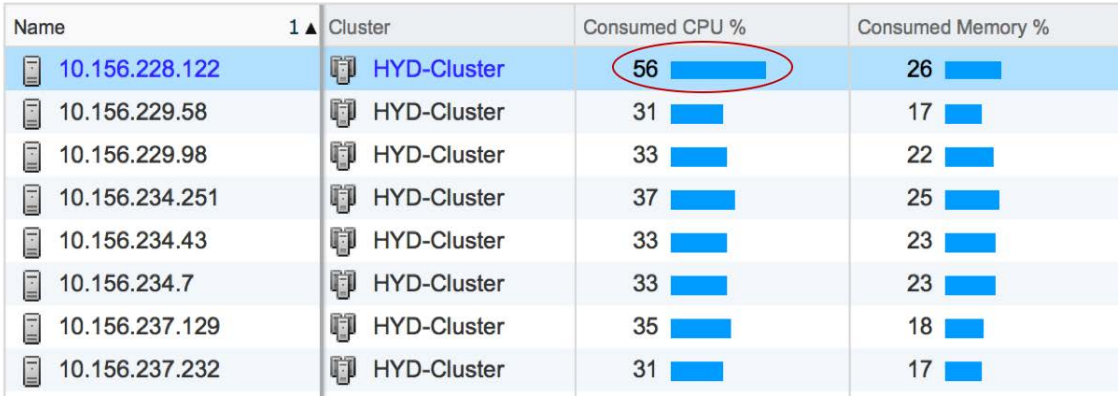
Figure 14 - CPU utilization view showing load unchanged in network saturated host after DRS migrations

Pair-Wise Balancing

DRS balances the cluster load using a standard deviation–based approach, where it always tries to bring the current host load standard deviation below a target value. This works well for cases where cluster sizes are moderate, but it does not always guarantee evenly distributed load across all hosts when the cluster size is large.

To illustrate this, we took eight hosts in a cluster (as shown in Figure 15), with seven of them being 30% utilized and one host being around 60% utilized. As you can see in the figure, the CPU usage of one host is an outlier. Though the utilization difference between the most utilized host and the least utilized one is fairly big, the load standard deviation across all the hosts is still relatively small. As a result, DRS chose not to make any migrations to further balance the cluster.

However, minimizing the difference between the maximum utilized host and minimum utilized host in such clusters may still be desirable (for example, for availability purpose, or for avoiding host resource over-commitment). To deal with such scenarios, DRS contains a new feature in vSphere 6.5 called pair-wise balancing, which is enabled by default.



Name	Cluster	Consumed CPU %	Consumed Memory %
10.156.228.122	HYD-Cluster	56	26
10.156.229.58	HYD-Cluster	31	17
10.156.229.98	HYD-Cluster	33	22
10.156.234.251	HYD-Cluster	37	25
10.156.234.43	HYD-Cluster	33	23
10.156.234.7	HYD-Cluster	33	23
10.156.237.129	HYD-Cluster	35	18
10.156.237.232	HYD-Cluster	31	17

Figure 15 - Example cluster, where one host's resource usage stands out from the rest

How It Works

DRS first distributes the load across hosts using the default standard deviation–based approach. Once the cluster is balanced, it checks if the resource usage of any host is an outlier. If it detects an outlier, DRS continues to move VMs in order to minimize the load difference between the most utilized and the least utilized host pair (or the max-min pair). Pair-wise balancing is explained with an example in the

Case Study below.

Things to remember:

- DRS tries to clear pair-wise (max-min pair) imbalance whenever possible, but doesn't guarantee to clear it.
- DRS tries to clear pair-wise imbalance only after the regular load balancing is completed.
- When you see DRS migrating VMs, even when the vSphere Web Client shows the cluster as balanced, it means DRS is trying to clear a pair-wise imbalance.
- Pair-wise balancing can be turned off by setting the cluster advanced option **CheckPairwiseImbalance** to 0.

Pair-Wise Balance Thresholds

The maximum allowed CPU or memory load difference between the most utilized and the least utilized host that pair-wise balancing can tolerate depends on the cluster **Migration Threshold** setting.

For the default Migration Threshold level of 3, the tolerable pair-wise balance threshold is 20%. That is, if there exist any two hosts in the cluster whose CPU or memory usage difference is more than 20%, pair-wise balancing will try to clear the difference by moving VMs from a highly utilized host to a lightly utilized host. The following table (Table 1) explains how much imbalance will be tolerated for different levels of migration threshold.

Migration Threshold level	Tolerable CPU/Memory usage difference between any two hosts in the cluster
1	NA
2	30%
3 (Default level)	20%
4	10%
5	5%

Table 1 - Summary of pair-wise imbalance tolerance for different levels of migration threshold

Case Study

To understand pair-wise balancing better, let us look at an example. We have a cluster with four hosts, where the load is fairly balanced.

As we can see from Figure 17, the difference between the CPU usage of hosts “10.156.234.43” and “10.156.229.98” is more than the tolerable threshold of 20%, with DRS migration threshold set to the default value. But you can see from Figure 16 that the cluster is balanced (per the standard deviation approach) because the load distribution is fair.

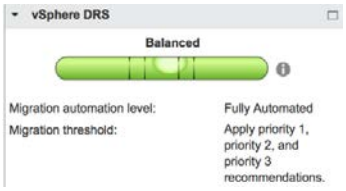


Figure 16 - Cluster balance view

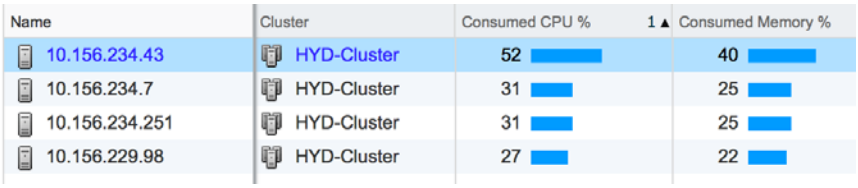


Figure 17 - Cluster load distribution summary view

At this point, pair-wise balancing recommends migrations to bring down the pair-wise difference between the hosts to less than the tolerable threshold of 20%. Figure 18 shows the list of migrations that happened and Figure 19 shows the pair-wise difference has come down to 17%.

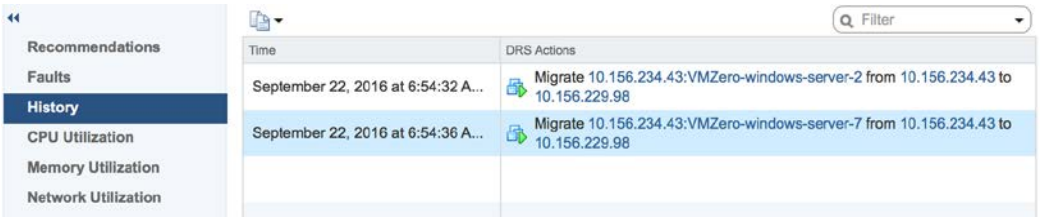


Figure 18 - DRS history view shows vMotions that DRS performed

Figure 19 - Cluster load distribution view shows reduction in pair-wise difference

New Additional Options

In order to simplify cluster management, in vSphere 6.5 three new options are exposed in the vSphere Web Client, called **Additional Options**. These options provide simple customizations to DRS behavior to better suit varied cluster needs. These additional options are pre-existing **advanced cluster options** with some predefined settings.

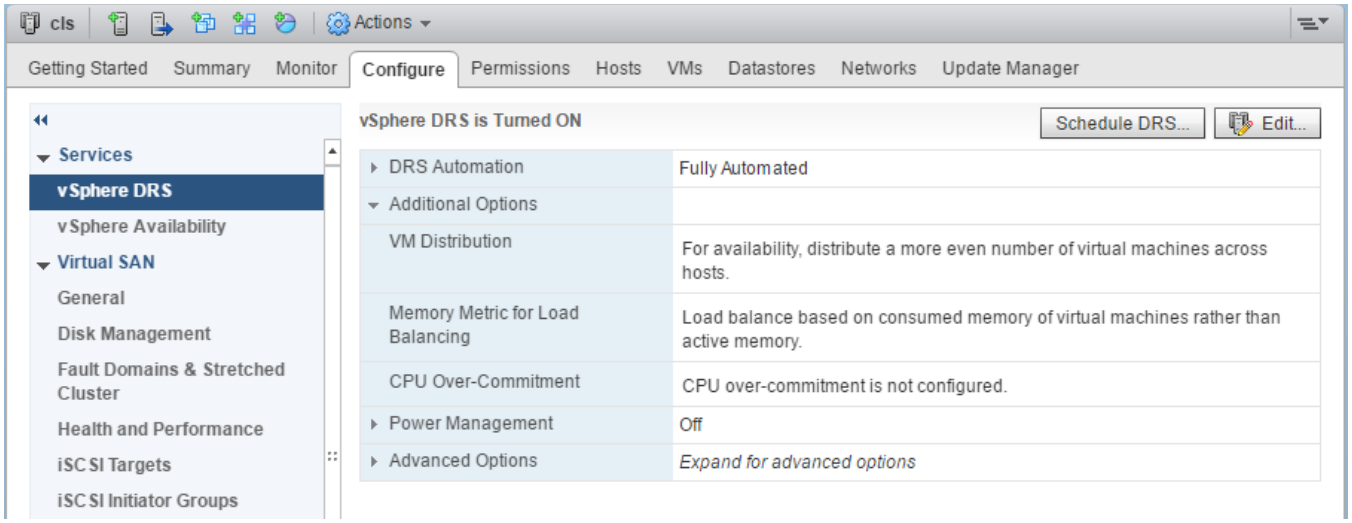


Figure 20 - New Additional Options for DRS in vSphere 6.5

These additional options are available, along with the existing configuration settings, in the **Cluster→Configure→vSphere DRS** in the vSphere Web Client (Figure 20).

VM Distribution

This additional option specifies DRS to consider distributing VMs evenly across hosts in the cluster for availability purposes. While the primary goal of DRS is to ensure that all VMs are getting the resources they need and that the load is balanced in the cluster, with this option, DRS will additionally try to ensure that the VM spread (number of VMs per host) is even across the cluster. However, DRS will always prioritize load balancing over the VM spread, so even distribution of VMs is done on a best-effort basis.

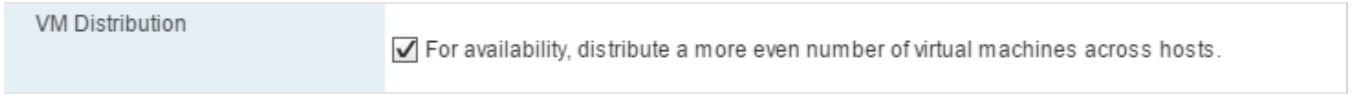


Figure 21 - Edit additional option for VM Distribution

You can enable this option by selecting the corresponding check box under **Additional Options** in the vSphere Web Client (Figure 21).

Memory Metric for Load Balancing

DRS, by default, mainly considers the active memory usage for load balancing [1]. This additional option makes DRS consider consumed memory usage, instead of active memory usage for load balancing. This option is equivalent to setting the existing cluster advanced option **PercentIdleMBInMemDemand** with a value of 100.

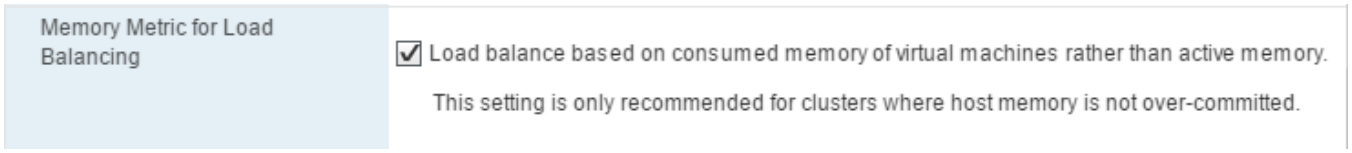


Figure 22 - Edit additional option for Memory Metric for Load Balancing

You can enable this option by selecting the corresponding check box under **Additional Options** in the vSphere Web Client (Figure 22).

CPU Over-Commitment

DRS supports over-committing physical resources for your VMs. This is very useful when you need to consolidate your workloads for better utilization of hardware resources. This additional option lets you specify the amount of CPU over-commitment as a percentage of total cluster CPU capacity that DRS should consider. You can specify a minimum of 0 and a maximum of 500% over-commitment ratio. This additional option is equivalent to using the pre-existing cluster advanced option **MaxClusterCpuOvercommitPct.**

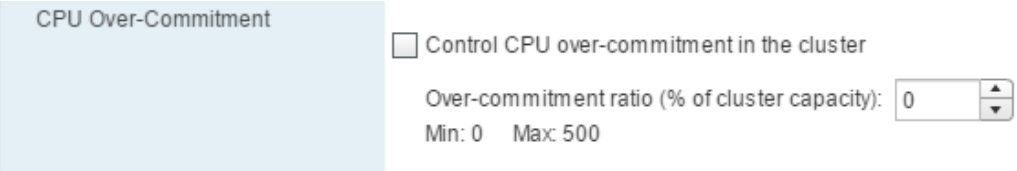


Figure 23 - Edit additional option for CPU Over-Commitment

You can enable this option by selecting the corresponding check box under **Additional Options** in the vSphere Web Client, and by specifying the over-commitment percentage (Figure 23).

The additional DRS options provide more power to you. They simplify cluster management with easy-to-use knobs in the vSphere Web Client.

Note that these additional options will override any equivalent cluster advanced options. For example, if you set cluster advanced option **PercentIdleMBInMemDemand** to some value, and then enable the memory metric option for load balancing, the advanced option will be cleared to give precedence to the new memory metric option.

Performance Improvements

All-New VM Initial Placement

In vSphere 6.5, DRS uses a completely new initial placement algorithm which is faster, lighter, and more effective than the previous algorithm.

In earlier versions, DRS took a snapshot of the cluster state to come up with a host recommendation for VM initial placement. In vSphere 6.5, the new algorithm completely avoids snapshotting, so generating recommendations for placement is much faster and the recommendations are also more accurate. Another advantage of avoiding snapshotting is that DRS can now support power-on operations with a much higher concurrency.

This new initial placement algorithm is not supported for certain configurations in vSphere 6.5. In cases where the new algorithm is not supported, DRS falls back to the previous snapshot-based algorithm to place VMs.

Configurations where the new initial placement algorithm is **not supported**:

- Clusters where DPM/Proactive HA/HA with admission control is enabled
- Clusters with DRS configured in manual mode
- VMs with manual DRS override setting
- VMs that are FT enabled
- VMs that are part of a vApp

To showcase the effectiveness of the new placement algorithm, we ran the following experiments in our lab using vSphere 6.5 in a cluster with only DRS enabled and compared the results with vSphere 6.0.

Sequential VM Placement

In the first experiment, we have 64 identical VMs and 4 identical, homogenous hosts in a DRS-enabled cluster. Initially, all of the 64 VMs are registered on the same host. We then issue multiple power-on requests for all VMs sequentially. We compare the placement result in vSphere 6.0 with that in vSphere 6.5.

Host	VMs before placement	VMs after placement	
		vSphere 6.0	vSphere 6.5
Host-1	64	21	16
Host-2	0	15	16
Host-3	0	14	16
Host-4	0	14	16

Table 2 - Placement of VMs during sequential power-on in vSphere 6.0 vs. vSphere 6.5

Concurrent VM Placement

Next, we take the same cluster, with all 64 VMs registered on the same host, and issue 64 concurrent power-on requests. Again, we compare the placement result in vSphere 6.0 with that in vSphere 6.5 running the new algorithm.

Host	VMs before placement	VMs after placement	
		vSphere 6.0	vSphere 6.5
Host-1	64	30	16
Host-2	0	16	16
Host-3	0	3	16
Host-4	0	15	16

Table 3 - Placement of VMs during concurrent power-on in vSphere 6.0 vs. vSphere 6.5

As we can see from the results in Table 2 and Table 3, the new VM initial placement mechanism in vSphere 6.5 places VMs more evenly upon power-on.

VM Power-On Latency

In order to showcase the overall performance improvements in VM placement in vSphere 6.5 (including the new placement algorithm), we studied the power-on operational latency when running our benchmarking tool against a vSphere lab test bed with 64 hosts and 8000 VMs in a DRS-enabled cluster.

The operations in the benchmark are described in section Benchmark Workload. To showcase the performance with increased concurrency, we ran two types of workload in our benchmark: light and heavy. The heavy workload had 4x the number of client threads as the light workload. We compared the median latency for a single VM power-on operation in these two workloads.

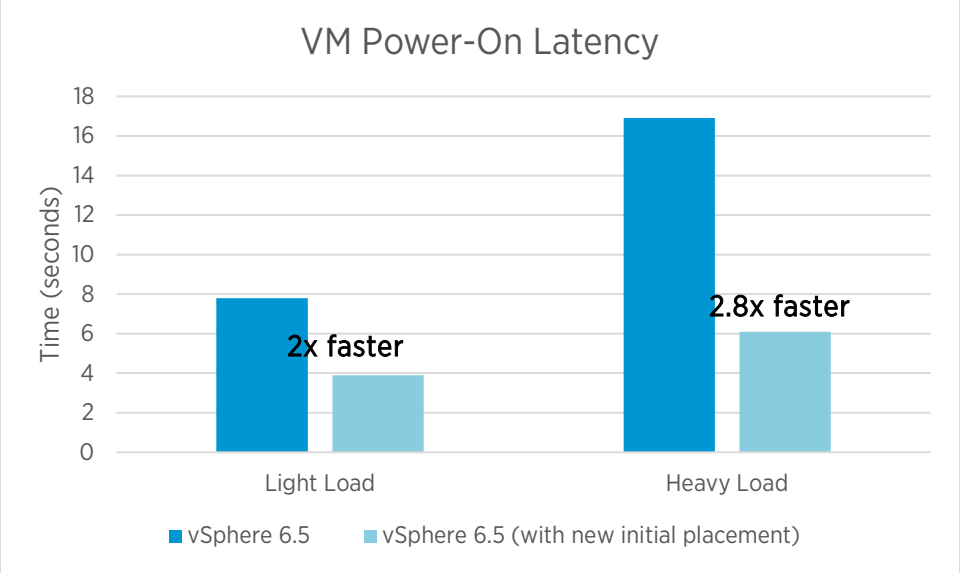


Figure 24 - Power-On latency improvements due to new placement algorithm

As we can see in Figure 24, the average VM power-on operation is generally much faster (over 2x) with the new initial placement algorithm. We can also see that it is even faster (2.8x vs. 2x) with a higher concurrency in workload.

The new initial placement algorithm is not only more effective in placing VMs, but also much faster and more efficient with higher concurrency.

Faster and Effective Maintenance Mode Operation

Evacuating a host during maintenance mode operation is now faster and more effective with DRS in vSphere 6.5. There are two major enhancements in this regard:

- Parallel evacuation of powered-off VMs
- Prioritizing the evacuation of VM templates and powered-off VMs over powered-on VMs

Parallel Evacuation of Powered-Off VMs

Evacuation of powered-off VMs is significantly faster in vSphere 6.5 with parallel evacuation. For every powered-off VM, DRS will pick the right host that can accommodate it. Once the destination hosts are chosen for all the powered off VMs on the host, the evacuation happens in parallel, moving the VMs all at once and re-registering them on their destination hosts.

By default, there can be a maximum of 100 VM re-register threads per vCenter Server that can be spawned at once to evacuate VMs. So, for up to 100 VMs, you may not see a significant change in evacuation times of powered-off VMs. Beyond 100 VMs, you will see increasing evacuation times (but still lower than in vSphere 6.0). Table 4 shows the comparison of evacuation times in vSphere 6.0 vs. vSphere 6.5 for the same number of VMs.

Number of powered-off VMs	Evacuation times (seconds)	
	vSphere 6.0	vSphere 6.5
64	40	8
128	89	14
256	171	30
512	334	58

Table 4 - Comparing evacuation times for powered-off VMs in vSphere 6.0 vs. 6.5

Prioritizing Evacuation of VM Templates and Powered-Off VMs

With many powered-on VMs in the host, maintenance mode generally takes a long time. Most of this time can be attributed to migrating live VMs. During this time, if you want to use any VM templates or power-on any powered-off VMs, you might have to wait for a non-trivial amount of time. Although migration of VM templates

and powered-off VMs to different hosts is much faster than live migrations, the VMs may end up waiting in a queue behind live migrations.

To avoid this, in vSphere 6.5, DRS prioritizes the migration of VM templates and powered-off VMs to other hosts, so that they can be accessible while the maintenance mode operation is still in progress.

Faster and Lighter DRS

In vSphere 6.5, DRS has been optimized for better resource usage and operational latency.

To showcase the performance improvements, we set up a testbed in our lab with the maximum supported configuration for a cluster: 64 ESXi hosts and 8000 VMs in a single DRS-enabled cluster. We ran an in-house benchmark that issues a mix of management operations to the vCenter Server via the vSphere Web Services SDK to simulate a high churn vCenter Server environment [2].

Details about the testbed and the software configurations used for this study can be found in the Experimental Setup for Performance section. The benchmark and the operations performed as part of this experiment can be found in the Benchmark Workload section. The benchmark reports the throughput that the server can drive, in number of operations per minute, and the overall latency of operations.

We then compared the data in the same testbed using vCenter Server 6.0 and 6.5.

Operational Latency

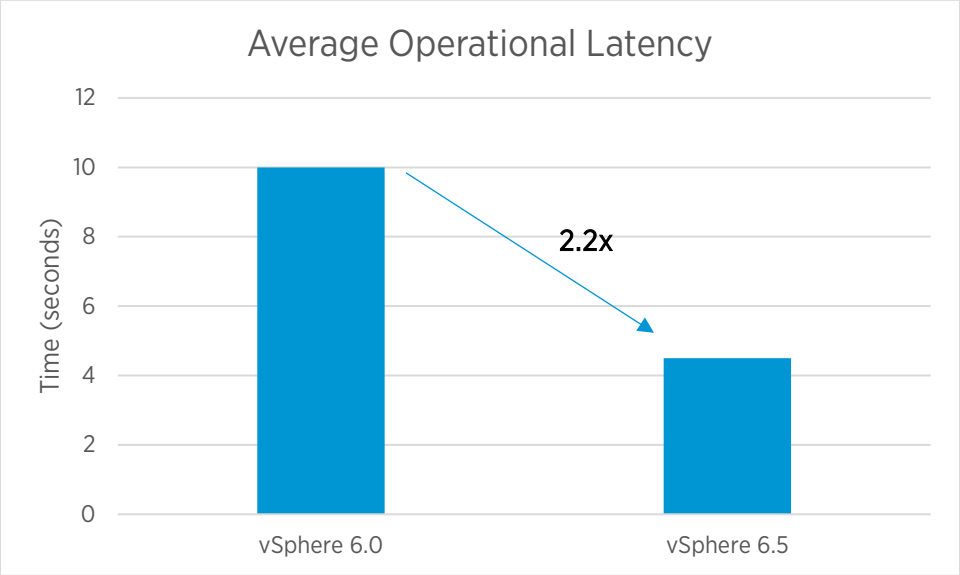


Figure 25 - Average overall operational latency in the performance benchmark

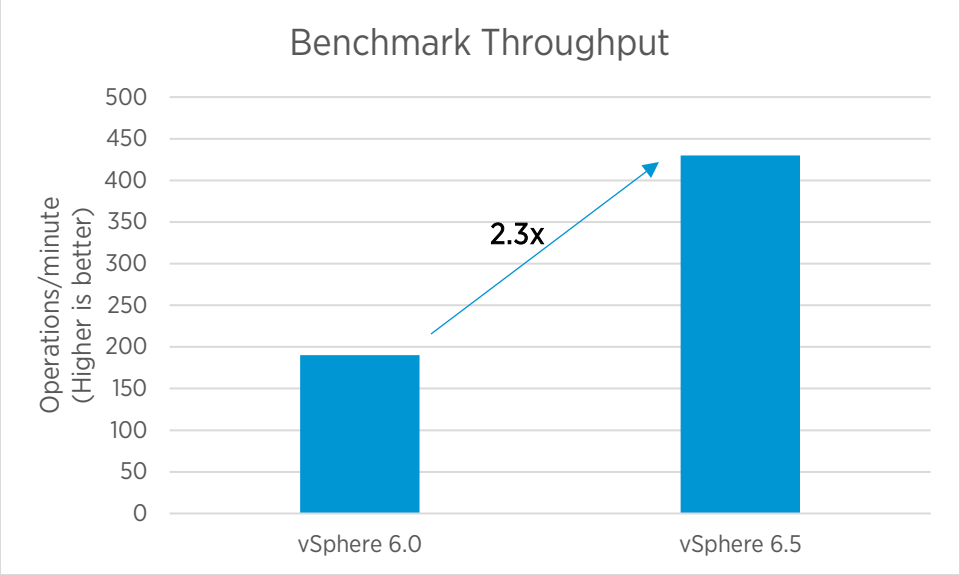


Figure 26 - Throughput in operations/min in the performance benchmark

The operations in the benchmark that take a significant amount of time are Power-on VM, vMotion VM, Group Power-On VMs, and Reconfigure VM. These operations significantly influence the overall throughput and latency of operations in the benchmark. The performance of these operations is largely determined by DRS, since they all rely on DRS to get recommendations. As a result, any performance improvements in the benchmark typically reflect improvements in DRS performance.

We noticed a 2.2x improvement in average operational latency (averaged across all the operations in the benchmark) in vSphere 6.5 (Figure 25).

We also noticed an improvement in throughput of 2.3x in vSphere 6.5 due to performance improvements in DRS (Figure 26).

vCenter Server Resource Usage

As part of the study, we also measured the resource usage of vCenter Server (which also includes DRS). In our vCenter Server, all the hosts and VMs in the inventory are placed in the single cluster that is being managed by DRS. Since this is the largest supported configuration for a DRS Cluster, the resource usage of vCenter Server in this configuration is a good indicator of DRS resource usage pattern in the worst case.

We measured the CPU and memory (resident set) usage of vCenter Server over the course of the benchmark run. We then computed the average for CPU usage and maximum for memory usage. Figure 27 shows the average CPU usage compared between versions 6.0 and 6.5. Figure 28 shows the maximum memory usage compared between 6.0 and 6.5.

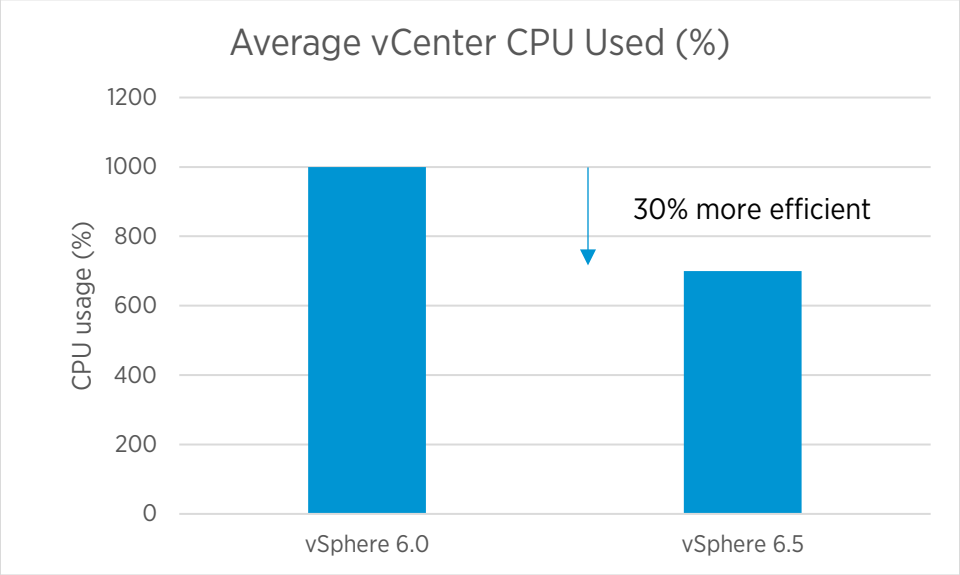


Figure 27 - Average vCenter Server CPU used (%), where one full CPU core represents 100%

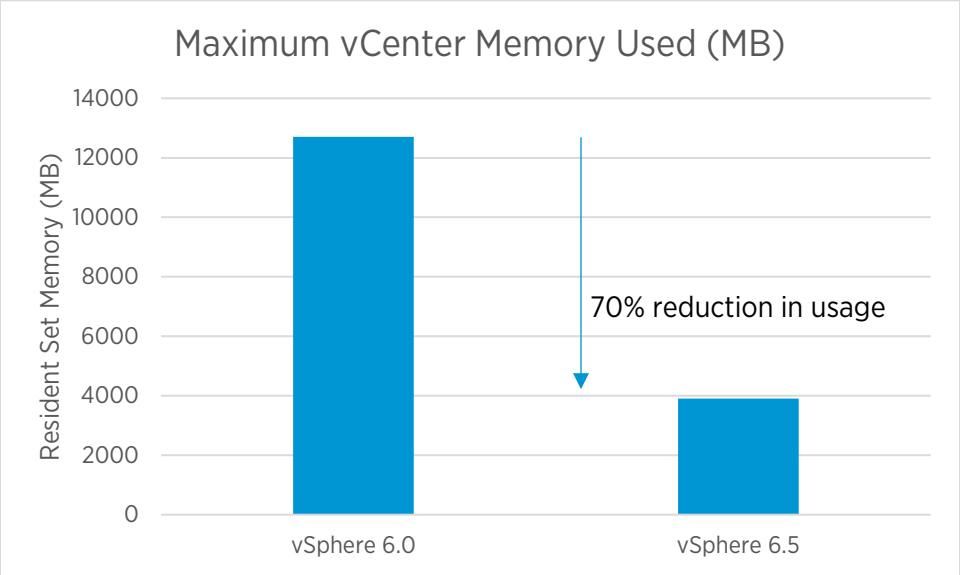


Figure 28 - Average vCenter Server memory used in MB

In vSphere 6.5, vCenter Server (including DRS) has been optimized to be 30% more efficient in CPU usage (Figure 27) and to use 70% less memory compared to vSphere 6.0 (Figure 28) for the same vCenter inventory size.

Conclusion

vSphere 6.5 brings a host of performance improvements and new features in DRS, allowing you to get the best performance out of your vSphere clusters. In this paper, we showcased some of these improvements and features with examples and gave some guidelines for best performance. We also showed how DRS has become faster, lighter, and more effective in the latest release.

References

[1] Vikas Madhusudana, Adarsh Jagadeeshwaran, and Sai Manohar Inabattini. (2016) Understanding vSphere DRS Performance.
http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vsphere6-drs-perf.pdf

[2] VMware, Inc. (2016) Performance Best Practices for VMware vSphere 6.0.
http://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/vmware-perfbest-practices-vsphere6-0-white-paper.pdf

[3] Sai Manohar Inabattini and Adarsh Jagadeeshwaran. (2015) VMware vCenter Server 6.0 Cluster Performance.
http://www.vmware.com/resources/techresources/10494

[4] VMware, Inc. (2015) vSphere 6 Resource Management Guide.
https://pubs.vmware.com/vsphere-60/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-60-resource-management-guide.pdf

[5] Aashish Parikh and Ajay Gulati. (2012) DRS: Advanced Concepts, Best Practices, and Future Directions.
http://download3.vmware.com/vmworld/2012/top10/vsp2825.pdf

[6] Ajay Gulati, Ganesha Shanmuganathan, Ann Holler, and others. (2012) VMware Distributed Resource Management Design, Implemenation, and Lessons Learned.
https://labs.vmware.com/vmtj/vmware-distributed-resource-management-design-implementation-and-lessons-learned

Appendix

Experimental Setup for Performance

Layout

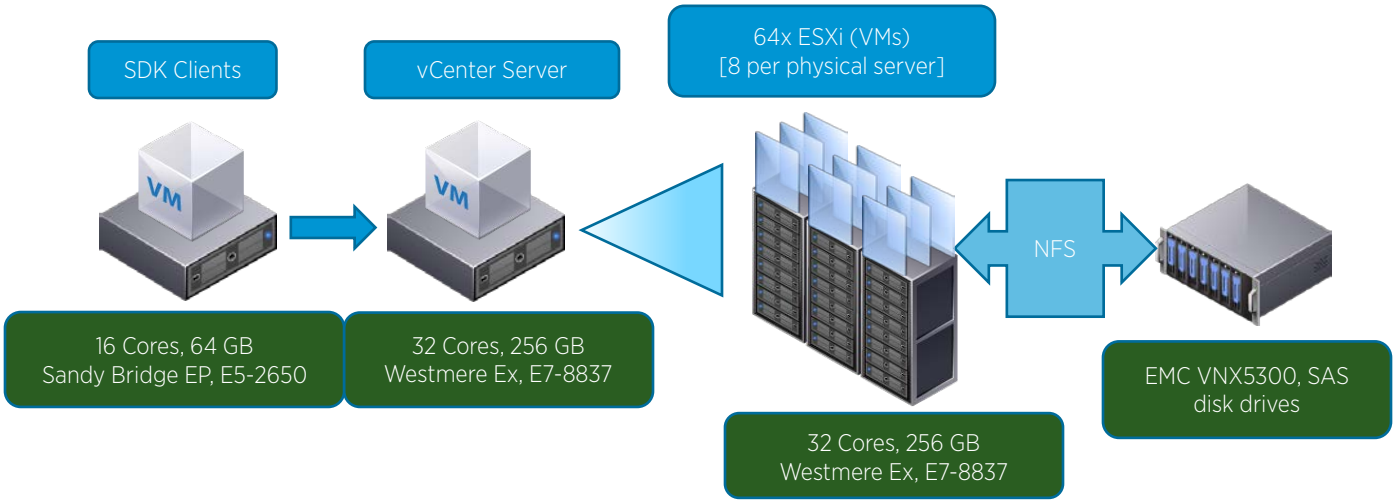


Figure 29 - Physical server layout in performance test bed

The cluster was configured with vSphere DRS and vSphere HA, and it contained a single root resource pool, with a set of resource pools under it, as shown in Figure 30.

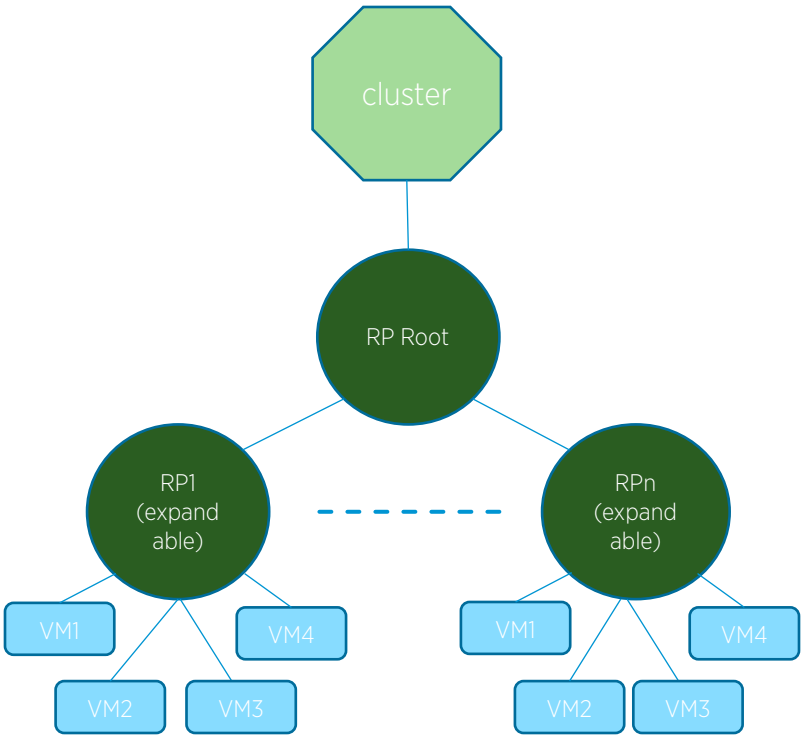


Figure 30 - Cluster Resource Pool structure in performance testbed

Software Configuration

vSphere Component	VMware Software Version	VM Configuration
ESXi Hosts	vSphere 5.5 U2 / vSphere 6.0	VM: 8 vCPUs, 32GB vRAM
vCenter Server Appliance (Linux)	vCenter Server 6.0	VM: 16 vCPUs, 32GB vRAM
	vCenter Serve 6.5	VM: 16 vCPUs, 32GB vRAM

Table 5 - Software configuration in the performance test bed

Benchmark Workload

The workload uses the vSphere Web Services SDK to simulate a high churn vCenter environment. Each client repeatedly issues a series of management and provisioning operations to vCenter Server. Table 6 lists the operations performed in this workload. These operations and their relative frequency were chosen from a sampling of representative customer data.

Operation	Description
Add Port Group	Create a new port group on a vSphere standard virtual switch
Remove Port Group	Remove a new port group on a vSphere standard virtual switch
Clone VM	Create a clone of a virtual machine.[1]
Create Folder	Create a new folder in the vCenter inventory hierarchy.
Delete Folder	Delete a folder from the vCenter inventory hierarchy.
Create Snapshot	Create a snapshot of a virtual machine.
Delete Snapshot	Delete a snapshot of a virtual machine.
Group Power-On VMs	Power-on several virtual machines in a single operation.
vMotion VM	Move a powered-on virtual machine to a different host.
Power-on VM	Power-on a single virtual machine in a DRS cluster.
Power Off VM	Power off a single virtual machine.
Reconfigure VM	Edit a virtual machine's configuration settings.[2]
Register VM	Add a .vmx file from a datastore to the vCenter inventory.
Unregister VM	Remove a virtual machine from the vCenter inventory without deleting its files from its datastore.
Relocate VM	Move a powered-off virtual machine to a different host.
Remove VM	Delete a single virtual machine from the vCenter inventory, and delete its files from its datastore.
Reset VM	Reset a single virtual machine.
Suspend VM	Suspend a single virtual machine.
Resume VM	Resume a single virtual machine.

Table 6 - List of management operations issued by performance benchmark

[1] The performance of a clone operation depends on the size of the VM being cloned.

[2] This benchmark's Reconfigure operation uses VM Memory Shares as a representative configuration setting.

About the Authors

Sai Manohar Inabattini is a senior performance engineer with the vCenter Server performance group. He works on vCenter Server performance and scalability, with special focus on the Distributed Resource Scheduling (DRS) algorithm. Sai holds a bachelor's degree in Computer Science from the National Institute of Technology at Warangal, India.

Vikas Madhusudana is a senior performance engineer with the vCenter Server performance group. He works on vCenter Server performance and scalability with special focus on development of tools for performance testing/automation. Vikas holds a bachelor's degree in computer science from the University Visvesvaraya College of Engineering, Bangalore, 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.

Acknowledgments

The authors would like to specially thank Fei Guo for his guidance and support for this paper. The authors would also like to thank Brian Graf, the DRS team, and the performance team for reviewing this 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

Comments on this document: https://communities.vmware.com/docs/DOC-32933