PERFORMANCE STUDY – APRIL 2017

# DRS CLUSTER MANAGEMENT WITH RESERVATION AND SHARES

vmware®

# Table of Contents



*NOTE: Parts of this document are taken with permission from VMware vSphere 5.1 Clustering Deepdive by Duncan Epping and Frank Denneman, pages 167-184 [1]. The information is still relevant to vSphere 6.0 and 6.5.*

# Reservation and shares - VM vs. resource pool

Reservation and shares are important resource management knobs in a vSphere DRS cluster. They can be set on cluster objects like VMs and resource pools to isolate resources, prioritize, and/or guarantee their availability. To know when to set them for VMs and when to set them on resource pools, we need to understand

- What these settings mean, and
- How these settings can impact resource availability for a VM

In this paper, we explain how these settings are different for a VM and resource pool while giving some general guidelines for using them.

## Reservation and shares for a VM

### Reservation

Reservation is a way of guaranteeing resources for a VM. When the user reserves resources for a VM, they are asking the host to make sure the VM always gets the reserved resources. The reservations set are static, which means that the VM is entitled to these resources whether it uses them or not. For this reason, resources reservations for a VM will have a multi-dimensional impact.

Resource reservation on a VM impacts

1. Resource reclamation – In case of contention for the resource, the host will reclaim resources from VMs, while still guaranteeing the minimum reserved resources.
2. Availability of resources for other VMs – Since resources are allocated to specific VMs with reservation, other VMs will have to share the remaining un-reserved resources.
3. VM placement and load balancing – During placement and load balancing, DRS also has to consider the reserved resources for VMs in the cluster, which sometimes leaves DRS with lesser options.

VM reservation works differently for memory and CPU. CPU instructions are transient and process context switching is typically very fast. So the CPU scheduler allows reserved, but unused vCPUs to be shared with other VMs. When it comes to memory, if it is shared with other VMs for temporary use, the data needs to be swapped out when the rightful owner wants to use this memory. Swapping existing data takes a significant amount of time, and that can delay the resource availability for the entitled VM, which is unfair. So the memory resources reserved for a VM are not shared with other VMs.

### Shares

Shares dictate the relative priority of resources for a VM, compared to resources of other child objects (siblings) of the same parent. The absolute number does not matter in shares, as the values are always relative. For example, if three VMs of the same resource pool have shares set to 16000, 8000 and 4000, the resources are distributed among the VMs in the ratio of 4:2:1 respectively.

Shares are allocated to each VM based on its configuration. By default, for every vCPU, 1000 shares, and for every MB of memory, 10 shares are allocated. For a VM with 2vCPUs and 4GB memory, 2000 CPU shares and 40960 memory shares are allocated (by default) by ESXi. This can be changed using the “resource settings” option for the VM.

ESXi considers these VM shares when scheduling unreserved resources. For every VM, ESXi computes something called the “ResourceUsagePerShare” value, which is equal to:

$$ResourceUsagePerShare = \frac{resource\ usage}{number\ of\ shares}$$

After all the VMs' resource reservations are met, if any VM requires more resources, then the VM with the lower ResourceUsagePerShare value will be scheduled first. So, if three VMs - VM1, VM2, and VM3 have shares 2000, 4000, and 8000 and have the same resource usage, then the VM with higher shares will be scheduled first, because its ResourceUsagePerShare value will be at the minimum. So the order of allocating unreserved resources will be VM3, VM2, and VM1.

This is the reason that when there is a contention in the host and all the VMs are competing for resources, VMs with high shares will get more resources. But, when resources are not over-committed and as long as resources are available in the host, every VM will get the amount of resources it needs.

## Reservation and shares for a resource pool

Now that we have seen how reservation and shares work for a VM, let's look into how they work for a resource pool and how they impact VMs' resource availability. Before we jump into the details, let's first understand what resource pools are, and why we need them.

There are two types of entities in a DRS cluster—resource consumers and resource providers. Hosts are resource providers and VMs are resource consumers.

It is important to realize that resource pools can be both consumers and providers of resources, as this might impact the way you design your resource pools from a shares perspective. Resource pools consume resources from hosts and provide resources to VMs ( Figure 1).

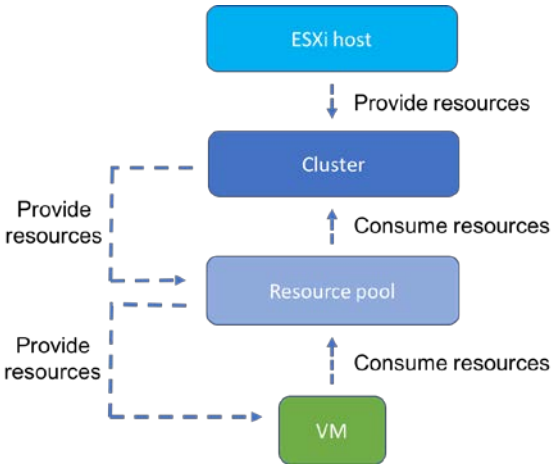


Figure 1. Providers and consumers of resources in a cluster

Resource pools are primarily used for sharing resources between identical VMs by pooling them together and isolating resources of different pools. This isolation can be achieved by using reservation and shares at the resource pool level. Resource pools are **not** folders. They should not be used just to group VMs. For grouping VMs, folders should be used, since for each resource pool created, DRS has to do some extra work of resource divvying.

When DRS is enabled in a cluster, a root resource pool will be created by default. All the hosts in the cluster will be part of this root resource pool. Child objects of a resource pool can be either resource pools or VMs.

### Shares

By default, a resource pool will be created with 4000 CPU shares and 163840 Memory shares. The resource entitlements for VMs are dependent on how the resource pool–level shares are distributed to individual VMs in the resource pool, which is also dependent on VM level shares. Let us take an example.

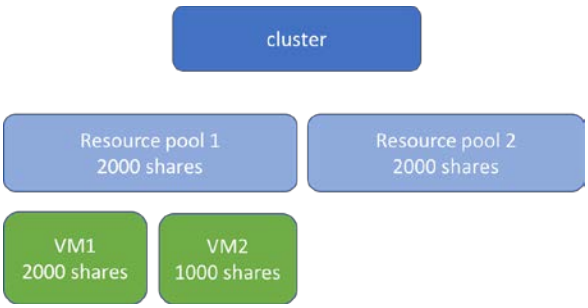


Figure 2. Example showing share distribution in a cluster

In the example shown in Figure 2, resource pools 1 and 2 have an equal amount of shares from the cluster. Which means they both are entitled to half the cluster resources each. Under resource pool 1, VM1 and VM2 have shares in the ratio 2:1. So VM1 is entitled to $^{2}/_{3}$ of resource pool 1 resources, and VM2 is entitled to $^{1}/_{3}$ of resource pool 1 resources. Hence the VM entitlements will be as follows.

$$Entitlement|VM1| = \frac{2}{3} \times \frac{1}{2}(cluster\ resource)$$

$$Entitlement|VM2| = \frac{1}{3} \times \frac{1}{2}(cluster\ resource)$$

When a VM and a resource pool are at the same level, they both compete for resources from the same parent. This would not be fair, since a resource pool contains multiple VMs, while the sibling VM consumes resources all for itself. Hence, it is not recommended to keep a VM and a resource pool at the same level.

## Resource pools and VMs as siblings

Having virtual machines as siblings to resource pools can affect the resource allocation of the subsequent layers. This section illustrates why placing virtual machines as siblings to resource pools is considered to be a misconfiguration. In the example in Figure 3, VM1 competes for resources with Resource pool 1. Based on their share values, each of them can receive up to 50% of the resources of the parent. The resources of Resource pool 1 need to be divided among its child-objects, contrary to VM1, which can use its allocated resources for its own sake.

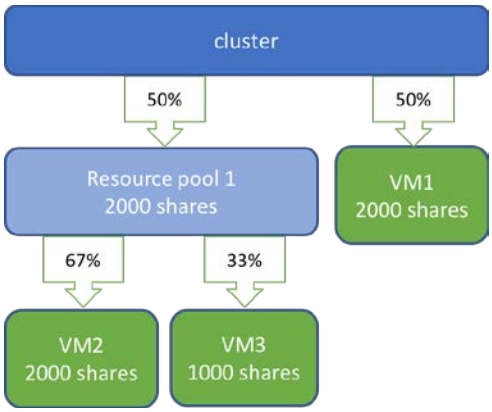


Figure 3. An example of resource pools and VMs as siblings

Two VMs (VM2 and VM3, with shares 2000 and 1000 each) are active inside Resource pool 1 and the resources are divided between those two VMs. Due to their VM share ratio (2:1), VM2 is entitled to 67%, and VM3 is entitled to 33% of the resources in Resource pool 1. When you add more VMs to the resource pool, the share value of each VM within that pool gets diluted, while the share value of VM1 remains the same, as you can see in Figure 4.

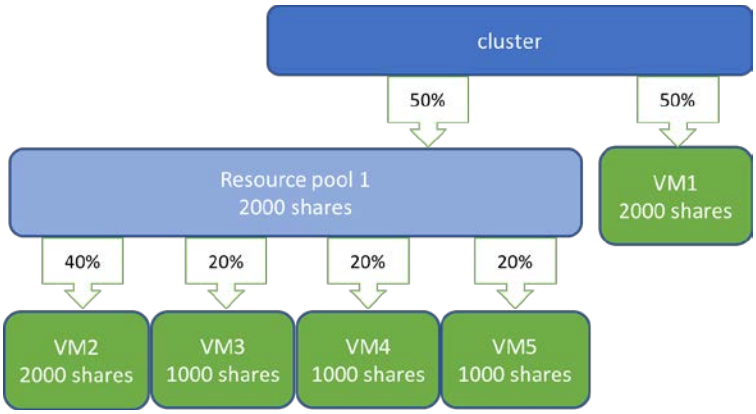


Figure 4. Example of VM shares getting diluted in a resource pool

Mixing virtual machines and resource pools at the same hierarchical level may create an unintended imbalance in the proportional share values at that hierarchical level and affects the resource allocation of subsequent layers.

## How do resource pool shares affect VM workloads?

DRS mirrors the resource pool hierarchy to each host and divides the entitled resources of the resource pool across the host-local resource pool tree based on the number of active virtual machines, their share amounts, and their current utilization. Once the resource allocation settings (reservation, limits, and shares) are propagated to the host local RP tree, the local host CPU and memory schedulers allocate resources based on those settings.

For example, a resource pool in a 2-host cluster is configured with a Normal CPU share level, as shown in Figure 5. Here, Resource pool 1 holds 4000 shares of CPU.

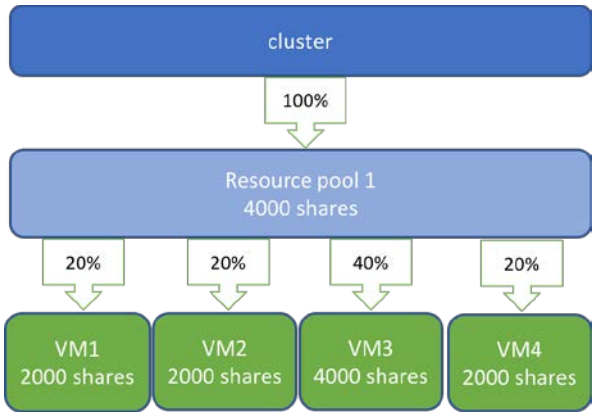


Figure 5. Single resource pool configuration

Four VMs running inside the resource pool can be configured as shown in Table 1.

| Virtual machine | Share level | Number of vCPUs | Shares | Share ratio | Host |
|---|---|---|---|---|---|
| **VM1** | Normal | 2 | 2000 | 2/10 | ESXi-01 |
| **VM2** | Normal | 2 | 2000 | 2/10 | ESXi-01 |
| **VM3** | High | 2 | 4000 | 4/10 | ESXi-02 |
| **VM4** | Normal | 2 | 2000 | 2/10 | ESXi-02 |

Table 1. Share configuration for VMs of Resource pool 1

Let's assume that all of the VMs are running the same, stable workload. DRS will balance the VMs across both hosts and will create the resource pool mapping shown in Figure 6.

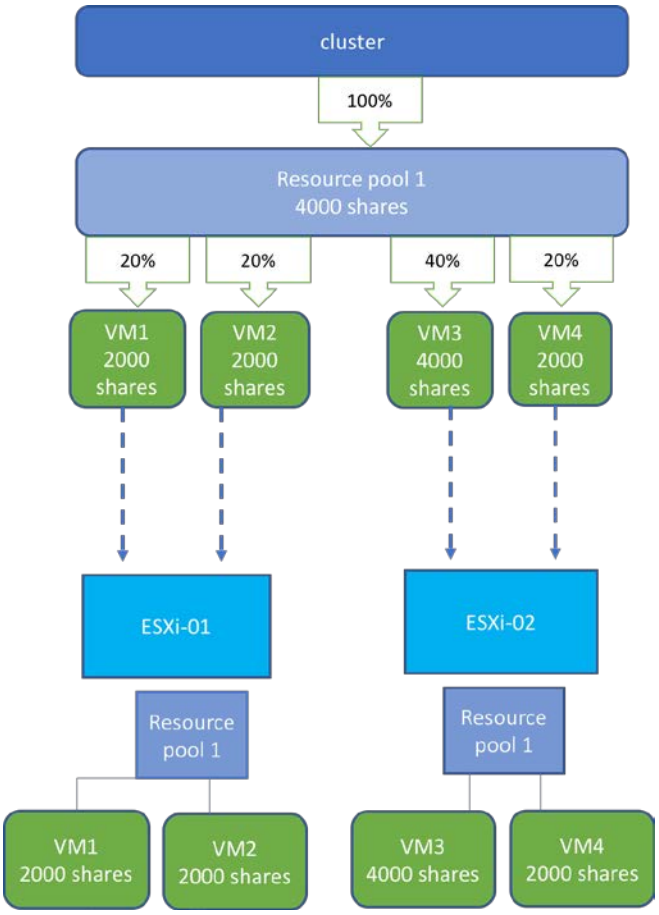


Figure 6. Host-local resource pool mapping

The number of shares specified on VMs VM1 and VM2 add up to 4000, which equates to 40% of the total configured VM shares inside the resource pool. In this example, DRS places VM1 and VM2 on ESXi-01 and for that reason assigns 1600 (40% of the total 4000) resource pool shares to Resource pool 1 of the host local RP tree. VM3 and VM4 are placed on ESXi-02 and receive the other 2400 (60% of 4000) of the resource pool shares.

At this point, we create Resource pool 2. Resource pool 2 is configured with a High share level and receives 8000 resource pool shares. The VMs of Resource pool 2 are configured identically to the VMs in Resource pool 1, as shown in Table 2.

| Virtual machine | Share level | Number of vCPUs | Shares | Share ratio | Host |
|---|---|---|---|---|---|
| **VM5** | Normal | 2 | 2000 | 2/10 | ESXi-01 |
| **VM6** | Normal | 2 | 2000 | 2/10 | ESXi-01 |
| **VM7** | High | 2 | 4000 | 4/10 | ESXi-02 |
| **VM8** | Normal | 2 | 2000 | 2/10 | ESXi-02 |

Table 2. Share configuration for VMs of Resource pool 2

The host-local resource pool tree of ESXi-01 is updated with Resource pool 2 and its VMs; Resource Pool 2 is configured with twice the number of shares as Resource pool 1. Introducing 3200 more shares from RP2 will increase the total number of resource pool shares in the host to 4800. Since Resource pool 2 owns 3200 of the total of 4800, it gets 66.6% of the host's resources.

Due to the ratio of 67:33 at the resource pool level, the local resource scheduler will allocate more resources to Resource pool 2 when contention occurs. The resources allocated to Resource pool 2 are distributed across the VMs based on their hierarchical level (sibling rivalry). This means that VM6 is entitled to 50% of Resource pool 2's resources during contention; this translates to 33% of the host's resources, as shown in Figure 7.

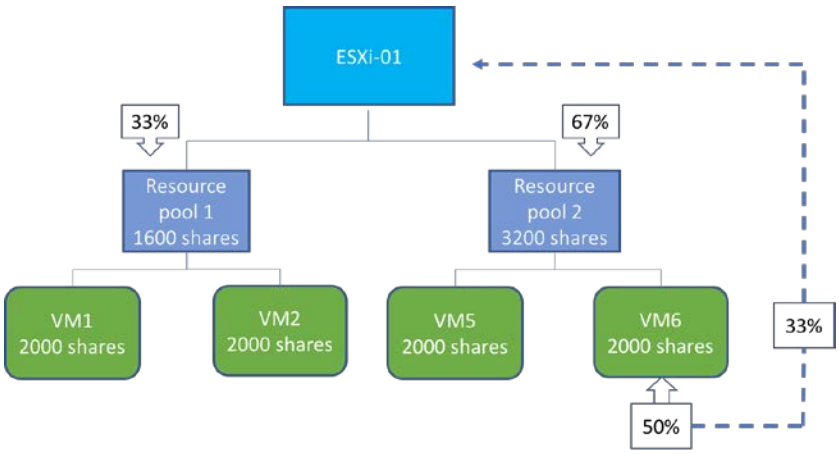


Figure 7. Resultant share ratios after adding a new resource pool with higher shares

## Reservation

A reservation on a resource pool is completely different from a reservation on a VM level. Reservations on a VM are static, and reserved resources will not be shared with other VMs. On the other hand, resource pool reservations are more dynamic, and unused resources will be shared with other resource pools.

Setting a reservation guarantees the permanent availability of physical resources to the resource pool. Because reservations are applied before shares, resources protected by a reservation are not reclaimed during contention. Setting a reservation at a resource pool–level ensures that the reserved resources are available for the resource pool and its child objects.

A reservation associated with a resource pool applies to all the VMs within the resource pool collectively. A resource pool–level reservation does not implicitly translate to a reservation for a child object, but rather "lends out" a portion of the protected resource. A resource pool reservation is divided among its children based on the dynamic entitlement of each child object. Dynamic entitlement is based on the configured size, resource allocation settings, demand, and the level of contention.

Because a resource pool reservation does not become a static setting on a child object, the amount of resources protected by the reservation can fluctuate for each child object from time to time.

By using the VM's dynamic entitlement, resource pool reservations have a dynamic nature and are more in line with the concept of consolidation and fairness. Reserved resources are divided among the VMs that require them; unused resources will be available, and ready to be used by other resource pools in the cluster.

For example, let us consider a cluster with two resource pools, RP-1 and RP-2. RP-1 has four VMs and a reservation of 10GB of memory. Now, based on the their dynamic entitlement, the 10GB reservation will be divided among the four VMs. For simplicity, let us assume that all four VMs are of the same size and have the same number of shares. So their dynamic entitlement will be based on the VMs' workload demand. The demands

of the four VMs are 1GB, 1GB, 2GB, and 4GB, respectively. The overall demand of the VMs put together is less than 10GB. So the total demand will be backed by a reservation and the remaining 2GB (10 - 8) can be used by the other resource pool, RP-2, as illustrated in Figure 8.

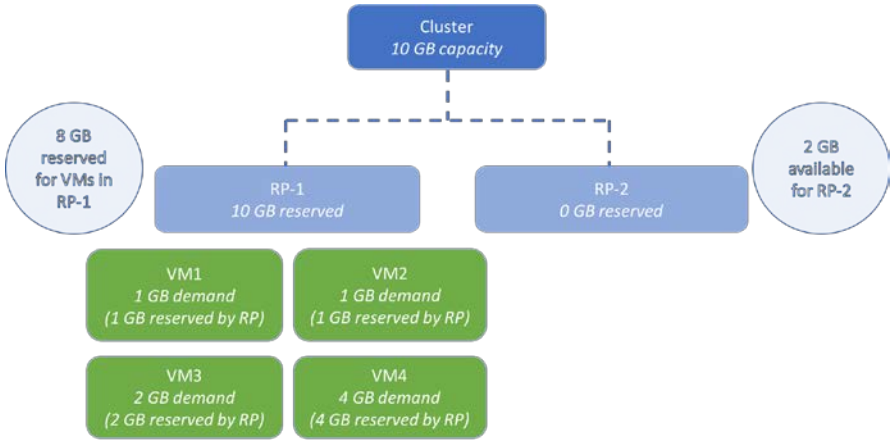


Figure 8. Resource pool-level reservation when overall VM demand is less than the amount reserved

When the demand for all the VMs increases to 4GB each, the resource pool reservation will be distributed among the VMs based on their shares, which in our case happens to be the same for all the VMs. Therefore the reservation will be distributed equally among them (2.5GB each), as illustrated in Figure 9.

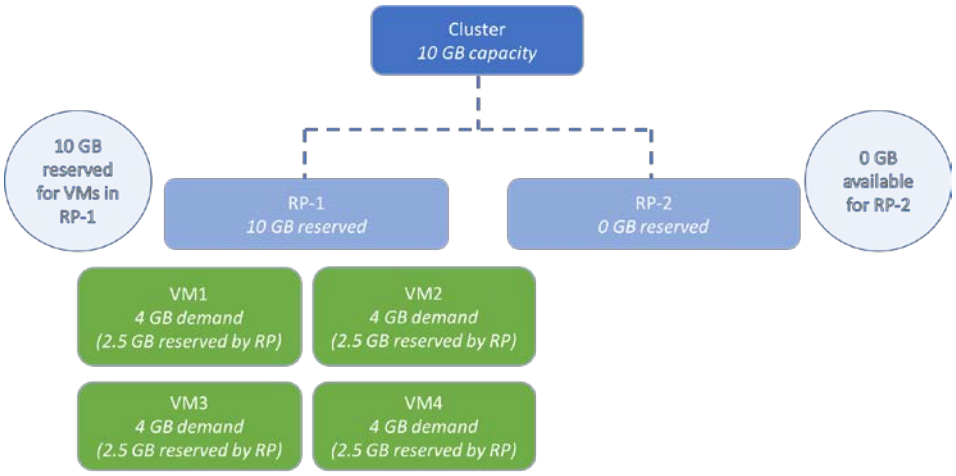


Figure 9. Resource pool–level reservation when the overall VM demand is more than the amount reserved

The dynamic nature of resource pool reservation might not be suitable for certain VMs: VM-level reservations are more suitable if the guaranteed availability of physical resources is required.

### VM-level reservation inside a resource pool

DRS passes VM-level reservations straight through to the host, where the host-level CPU and memory schedulers enforce the reservation. Any VM-level reservation is withdrawn from the resource pool–level reservation amount and reduces the amount of reserved resources available to its siblings. Physical resources allocated by the VM reservation are available only for that VM and will not be shared with siblings or VMs and resource pools external to the parent resource pool.

# Case Study: Reservation and shares at the resource pool–level

To help understand these concepts better, let us consider some examples.

Consider a case where we have three classes of VMs based on the service level agreement (SLA) that the applications on these VMs have to meet. We have a cluster with two ESXi hosts of 16GB memory each. For efficient use of resources, we would like to have all three classes of VMs in the same cluster.

Let's name the three classes of VMs A, B, and C:

- Class A – VMs running critical applications and cannot tolerate any shortage of resources.
- Class B – VMs running important applications and can tolerate a slight shortage of resources.
- Class C – VMs that are neither critical, nor important. These VMs can tolerate a shortage of resources.

For simplicity, let's assume that all the VMs are of same size and are configured with "normal" share values. Since the default value for VM shares depends on the size of the VM, and all VMs are of the same size in this case, they will all have the same number of shares.

Each VM is configured with 4GB and 1 vCPU.

With different categories of VMs together in a cluster, it's always a good practice to place them in different resource pools and use resource pool reservation and shares to ensure resource availability. Let us consider the resource pool hierarchy, as shown in Figure 10.

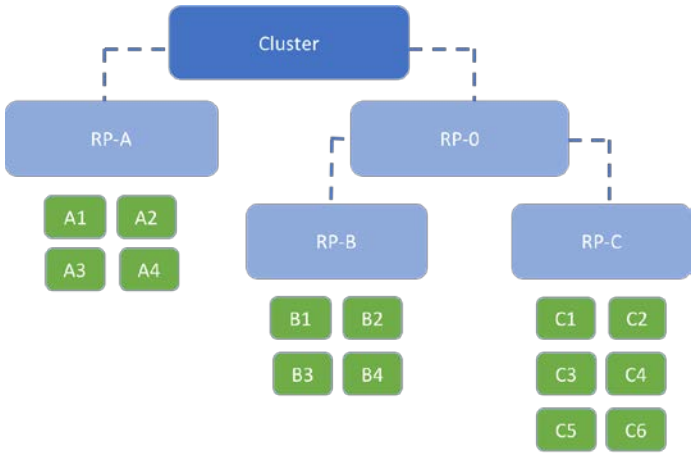


Figure 10. Resource pool hierarchy, case study 1

As we can see from Figure 10, there are 14 VMs in total in the cluster, and each has a VM demand of 4GB memory (100% demand). So there is a need for 56GB memory, while the cluster has only 32GB (2 hosts × 16GB each). Clearly, there is resource contention in the cluster, due to over-commitment.

Let's set the following resource settings for the resource pools, to prioritize resources based on VM classes and their importance:

- RP-A – 100% reservation for all the VMs' memory
- RP-B – Higher shares over RP-C

Since the absolute value of shares doesn't matter, for simplicity, let's assume that there is a total of 2400 shares at the cluster level.

So, with these settings, the resource pool hierarchy would look like that shown in Figure 11.

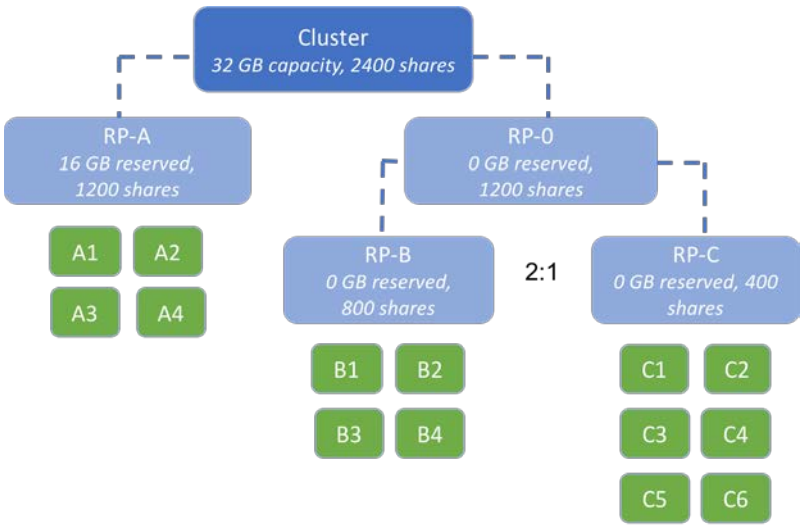


Figure 11. Resource pool hierarchy with reservation and shares settings

Now, the resources inside the resource pool are distributed based on VM shares and demand. Since all the VMs have the same amount of shares (as mentioned earlier), and our assumption is that every VM is asking for 100% of its configured resources, the demand of all the VMs will be the same.

So, inside the RP, resources will be distributed equally between the VMs:

- For Class A VMs: Since a reservation is set for the RP, reserved memory will be distributed among the VMs based on VM shares and demand. Since both shares and demand are the same for all VMs, reserved memory will be equally distributed.

  This means that each VM is entitled to 4GB of memory.
- For Class B VMs: Each VM is entitled to resources worth 200 shares (800 ÷ 4)
- For Class C VMs: Each VM is entitled to resources worth 66.6 shares (400 ÷ 6)

Now let's look into how the host will allocate resources based on this resource pool structure.

By default, any resource pool created will span all hosts. With DRS enabled, let's say that the VMs are distributed as shown in Table 3.

| Host | VMs |
|---|---|
| Host-1 | (A1, A2) (B1, B2) (C1, C2) |
| Host-2 | (A3, A4) (B3, B4) (C3, C4, C5, C6) |

Table 3. Distribution of VMs across hosts - case study 1

Once DRS constructs the resource pool structure, it pushes the resource pool structure into individual hosts to allocate resources, so the host resource pool structure will now look like the one shown in Figure 12.

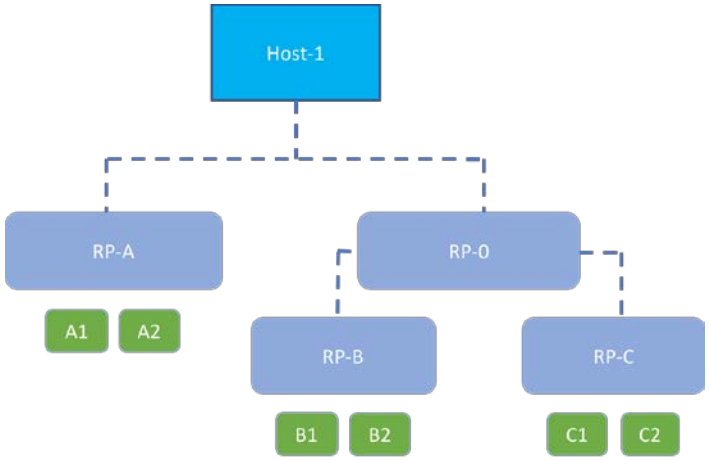


Figure 12. Host-1 resource pool structure, case study 1

In host-1, the resource distribution will be as shown in Table 4.

| | |
|---|---|
| **RP-A** | 2 (VMs) × 4GB = 8GB |
| **RP-B** | 2 (VMs) × 200 shares = 400 shares worth resources |
| **RP-C** | 2 (VMs) × 66.6 shares = 133.3 shares worth resources |

Table 4. Resource distribution on host-1, case study 1

During resource scheduling, ESXi will first allocate reserved resources to the entities (VMs/RPs) and then share the remaining resources based on entity (VM/RP) shares. In our case, since RP-A has already consumed 8GB of memory, which is equal to the total configured size of all VMs, no further resources will be needed for RP-A. The remaining 8GB will be entirely shared between RP-B and RP-C, based on their shares.

The ratio of RP-B and RP-C shares is 400:133 = 3:1

Therefore, the resource allocations in host-1 will be as follows:

- RP-A = 8GB
- RP-B = 6GB
- RP-C = 2GB

In host-2, the resource distribution will be as shown in Table 5.

| | |
|---|---|
| RP-A | 2 (VMs) × 4GB = 8GB |
| RP-B | 2 (VMs) × 200 shares = 400 shares worth resources |
| RP-C | 4 (VMs) × 66.6 shares = 266.6 shares worth resources |

Table 5. Resource distribution on host-2, case study 1

The ratio of RP-B and RP-C shares is 400:266.6 = 3:2

Therefore, the resource allocations in host-2 will be as follows:

- RP-A = 8GB
- RP-B = 4.8 GB
- RP-C = 3.2 GB

Let's now look at the resource demand vs. availability for VMs; this tells us how much resource each VM is getting, compared to what it needs. As we can see from Table 6 for host-1 and Table 7 for host-2, resources are clearly prioritized for VMs based on their class (priority).

| VM Type | Demand | Available |
|---|---|---|
| A | 4GB | 4GB |
| B | 4GB | 3GB |
| C | 4GB | 1GB |

Table 6. Demand vs. availability of resources for each VM on host-1, case study 1

| VM Type | Demand | Available |
|---|---|---|
| A | 4GB | 4GB |
| B | 4GB | 2.4GB |
| C | 4GB | 0.8GB |

Table 7. Demand vs. availability of resources for each VM on host-2, case study 1

## Takeaways

- Reservations at the resource pool–level can guarantee resource availability for all VMs in the resource pool, collectively.
- Shares can be used to prioritize resource availability for VMs at the time of contention.
- Adding more VMs to a resource pool can dilute the availability of resources for each VM in the pool.

# Summary

In this paper, we learned about two key resource management knobs in DRS—reservation and shares. With examples, we learned how, by setting reservations and shares in different ways, we can impact the availability of resources for a VM in the cluster.

# Frequently Asked Questions

**Q: When do I use VM reservations?**

A: When you need guaranteed resources for the individual VMs, you can use VM reservations. Remember, having too many VMs with reservations would reduce the available resources for unreserved VMs.

**Q: When do I use resource pool reservations?**

A: If there is a group of similar VMs that need guaranteed resources for the group, resource pool reservations will be helpful. Nothing is guaranteed for an individual VM, but only for the group. Most of the time, if the VMs in the resource pool are similar in terms of configuration and shares, the resource pool reservation will be distributed to the VMs based on their workload demand. So, if the VMs are not critical, it is advisable to use resource pool resevations for the group of similar VMs. Also, with resource pool reservations, the reserved and unused resources will be shared with other resource pools in the cluster, which is not the case with VM-level reservations.

**Q: When do I use VM shares?**

A: Shares are only used to prioritize resources. They do not guarantee resources. If you have multiple VMs of different sizes and workloads, you can use shares to prioritize resources for the time of resource contention. Remember, on a given host, in a given resource pool, a VM with higher shares will get more resources than other VMs in the same resource pool and host. Absolute values of resources for the VMs will depend on the resource availability on the host and the priorty for other resource pools on the same host.

# References

[1] D. Epping and F. Denneman, "vSphere 5.1 Clustering Deepdive," in *vSphere 5.1 Clustering Deepdive*, San Bernardino, CA, CreateSpace, 2013, pp. 167-184.

## 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.

**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.

## Acknowledgements

The authors would like to thank Fei Guo, Lan Gao, and Sabareesh Subramaniam from the vSphere Resource Management team, and Julie Brodeur from the Performance team for their help in reviewing the paper.

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