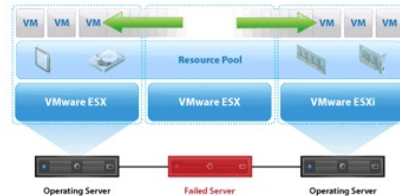


## HA Deepdive

My posts on VMware High Availability(HA) has “historically” been my best read post. I decided to rewrite the posts and create a page which is easier to maintain when functionality changes and a lot easier to find via Google and or the menu. This section is always under construction so check it every once in a while!



Everybody probably knows the basics of VMware HA so I'm not going to explain how to set it up or that it uses a heartbeat for monitoring outages or isolation. If you want to know more about the basics I recommend reading the [availability guide](#). Keep in mind that a lot of this info is derived from the availability guide, VMworld presentations and by diving into HA in my lab. I used these source articles as a basis for mine and tried to simplify the concepts to make it easier to digest.

I've divided the article into several sections and each section will contain a “basic design principle”:

- Node Types
- Node role
- Isolation response
  - Isolation response gotchas
- Admission control
- Host failures
  - What's a slot?
  - How does HA calculate how many slots are available per host?
- Percentage of cluster resources reserved
- Specify a failover host
- My Admission Control Policy Recommendation
- Flattening shares
- Advanced settings

## Node Types

A VMware HA Cluster consists of nodes, primary and secondary nodes. Primary nodes hold cluster settings and all “node states” which are synchronized between primaries. Node states hold for instance resource usage information. In case that vCenter is not available the primary nodes will have a rough estimate of the resource occupation and can take this into account when a fail-over needs to occur. Secondary nodes send their state info to the primary nodes.

Nodes send a heartbeat to each other, which is the mechanism to detect possible outages. Primary nodes send heartbeats to primary nodes and secondary nodes. Secondary nodes send their heartbeats to primary nodes only. Nodes send out these heartbeats every second by default. However this is a changeable value: `das.failedetectioninterval`. (Advanced Settings on your HA-Cluster)

The first 5 hosts that join the VMware HA cluster are automatically selected as primary nodes. All the others are automatically selected as secondary nodes. When you do a reconfigure for HA the primary nodes and secondary nodes are selected again, this is at random. The vCenter client does not show which host is a primary and which is not. This however can be revealed from the Service Console:

```
cat /var/log/vmware/aam/aam_config_util_listnodes.log
```

Another method of showing the primary nodes is shown in the following screenshot where `/opt/vmware/aam/bin/Cli` is used:


### ABOUT YELLOW-BRICKS.COM


Welcome to Yellow-Bricks.com a blog about virtualization, with a strong focus on VMware, HA and vCloud solutions.


Some facts:


1003 Blog Articles  
1998 Twitter Followers  
5593 RSS Subscribers

### SPONSORS









### ADVERTISEMENTS


**VMware Performance Issues**  
Virtual Wisdom is the answer. Resolve problems in real time  
[www.virtualinstruments.com](http://www.virtualinstruments.com)

**High Availability**  
Ensure High Availability For Your Infrastructure. Free Whitepaper!  
[www.progress.com/sonic](http://www.progress.com/sonic)

**VMware® Official Site**  
Increase Productivity & Reduce Your Costs by up to 60% with VMware®!  
[www.VMware.com](http://www.VMware.com)



### RECOMMENDED



**vSphere 4.0 Quick Start Guide**

Shortcuts down the path of Virtualization

Buy it now just \$15.99.

### ADVERTISEMENTS

**High Availability**  
Ensure High Availability For Your Architecture. Get The Whitepaper!

```

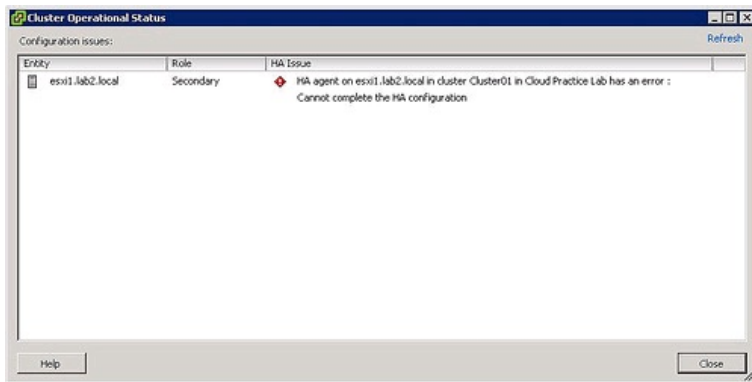
192.168.101.1 - PuTTY
/opt/vmware/aam/bin # ./Cli

Welcome to VMware AAM
Version 5.1.2
Copyright VMware, Inc. 2006

AAM> ln
*** Node esxi3 is the master primary ***
Node      Type      State
-----
esxi1     Primary   Agent Running
esxi2     Primary   Agent Running
esxi3     Primary   Agent Running
esxi4     Primary   Agent Running
esxi5     Primary   Agent Running
AAM>

```

As of vSphere 4.1 a third option has been added. It must be noted that this option will only show results when there is an error. If an error occurs you can easily check what the issue is by going to your cluster and clicking the "Cluster Operational Issues" line on the Summary tab. If there are no issues the screen will be completely gray. I forced an issue though so you can see what is shown.



Now that you've seen that it is possible that you can list all node with the CLI you probably wonder what else is possible... Lets start with a warning, this is not supported. Also keep in mind that the supported limit of primaries is 5, I repeat 5. This is a soft limit, so you can manually add a 6th, but this is not supported. Now here's the magic ...

Using Cli you can also promote nodes from secondary to primary and vice versa. This is shown in the following screenshots:

To promote a node:

```

192.168.101.1 - PuTTY
/opt/vmware/aam/bin # ./Cli

Welcome to VMware AAM
Version 5.1.2
Copyright VMware, Inc. 2006

AAM> demoteNode esxi1
OK
AAM>

```

To demote a node:

```

192.168.101.1 - PuTTY
/opt/vmware/aam/bin # ./Cli

Welcome to VMware AAM
Version 5.1.2
Copyright VMware, Inc. 2006

AAM> promoteNode esxi1
OK
AAM>

```

I can't say this enough, it is unsupported but it does work. With vSphere 4.1 a new advanced setting has been introduced. This setting is not even experimental, it is also unsupported. I don't recommend anyone using this in a production environment, if you do want to play around with it use your test environment. Here it is:

```

das.preferredPrimaries = hostname1 hostname2 hostname3
or
das.preferredPrimaries = 192.168.1.1,192.168.1.2,192.168.1.3

```

The list of hosts that are preferred as primary can either be space or comma separated. You don't need to specify 5 hosts, you can specify any number of hosts. If you specify 5 and all 5 are available they will be the primary nodes in your cluster. If you specify more than 5, the first 5 of your list will become primary.

When primaries are selected by using "promoteNode" or by powering up hosts in the right order you will need to verify occasionally if your hosts are still primary or not as HA has a re-election mechanism.

**Election time!**

Get the VMware paper:  
[www.progress.com/sonic](http://www.progress.com/sonic)

### Increase VMware use

If performance is a concern VirtualWisdom can help  
[www.virtualinstruments.com](http://www.virtualinstruments.com)

### VMware® Official Site

Increase Productivity & Reduce Your Costs by up to 60% with VMware®!  
[www.VMware.com](http://www.VMware.com)



Ads by Google

### RECENT COMMENTS

- ▶ [Artur on vCenter Resiliency?](#)
- ▶ [PiroNet on vCenter Resiliency?](#)
- ▶ [Tom on Standby vCenter Server for disaster recovery](#)
- ▶ [Tom on Standby vCenter Server for disaster recovery](#)
- ▶ [aosmak on vCenter Resiliency?](#)

### TAGS

[2.5 3.0.x 3.5 BC-DR Blogging Book](#)  
[Bugs cloud design drs ESX esxi ha](#)  
[Howto kb linux networking News patches](#)  
[performance powershell Scripting](#)  
[scripts security service console software srm](#)  
[Storage Tools training u2 update](#)  
[upgrade VCB vcdx vcenter vdi](#)  
[VirtualCenter vmtn VMware](#)  
[vmworld vSphere vstorage whitepaper](#)  
[Workstation](#)

### DISCLAIMER

The views expressed anywhere on this site are strictly mine and not the opinions and views of VMware.

Now when does a re-election occur? It is a common misconception that a re-election occurs when a primary node fails. This is not the case. The promotion of a secondary host only occurs when a primary host is either put in "Maintenance Mode", disconnected from the cluster, removed from the cluster or when you do a reconfigure for HA.

If all primary hosts fail simultaneously no HA initiated restart of the VMs will take place. HA needs at least one primary host to restart VMs. This is why you can only take four host failures in account when configuring the "host failures" HA admission control policy. (Remember 5 primaries...)

However, when you select the "Percentage" admission control policy you can set it to 50% even when you have 32 hosts in a cluster. That means that the amount of failover capacity being reserved equals 16 hosts.

Although this is fully supported but there is a caveat of course. The amount of primary nodes is still limited to five. Even if you have the ability to reserve over 5 hosts as spare capacity that does not guarantee a restart. If, for what ever reason, half of your 32 hosts cluster fails and those 5 primaries happen to be part of the failed hosts your VMs will not restart. (One of the primary nodes coordinates the fail-over!) Although the "percentage" option enables you to save additional spare capacity there's always the chance all primaries fail.

*Basic design principle: In blade environments, divide hosts over all blade chassis and never exceed four hosts per chassis to avoid having all primary nodes in a single chassis.*

## Node Role

You will need at least one primary because the "fail-over coordinator" role will be assigned to this primary, this role is also described as "active primary". I will use "fail-over coordinator" for now. The fail-over coordinator coordinates the restart of VMs on the remaining primary and secondary hosts. The coordinator takes restart priorities in account. Keep in mind, when two hosts fail at the same time it will handle the restart sequentially. In other words, restart the VMs of the first failed host (taking restart priorities in account) and then restart the VMs of the host that failed as second (again taking restart priorities in account). If the fail-over coordinator fails one of the other primaries will take over.

## Isolation Response

Talking about HA initiated fail-overs; one of the settings everyone has looked into is the "isolation response". The isolation response refers to the action that HA takes when the heartbeat network is isolated. Today there are three isolation responses, "power off", "leave powered on" and "shut down".

Up to ESX 3.5 U2 / vCenter 2.5U2 the default isolation response when creating a new cluster was "Power off". As of ESX 3.5 U3 / vCenter 2.5 U3 the default isolation response is "leave powered on". For vSphere ESX / vCenter 4.0 this has been changed to "Shut down". Keep this in mind when installing a new environment, you might want to change the default depending on customer requirements.

*Power off – When a network isolation occurs all VMs are powered off. It is a hard stop.*

*Shut down – When a network isolation occurs all VMs running on that host are shut down via VMware Tools. If this is not successful within 5 minutes a "power off" will be executed.*

*Leave powered on – When a network isolation occurs on the host the state of the VMs remains unchanged.*

The question remains, which setting should I use? It depends. I personally prefer "Shut down" because I do not want to use a deprecated host and it will shut down your VMs in a clean manner. Many people prefer to use "Leave powered on" because it reduces the chances of a false positive. A false positive in this case is an isolated heartbeat network but a non-isolated VM network and a non-isolated iSCSI / NFS network.

I guess most of you would like to know how HA knows if the host is isolated or completely unavailable when you have selected "leave powered on".

HA actually does not know the difference. HA will try to restart the affected VMs in both cases. When the host has failed a restart will take place, but if a host is merely isolated the non-isolated hosts will not be able to restart the affected VMs. This is because of the VMX file lock; no other host will be able to boot a VM when the files are locked. When a host fails this lock starves and a restart can occur.

The amount of retries is configurable as of vCenter 2.5 U4 with the advanced option "das.maxvmrestartcount". The default value is 5. Pre vCenter 2.5 U4 HA would keep retrying forever which could lead to serious problems as described in the [KB article](#).

The isolation response is a setting that needs to be taken into account when you create your design. For instance when using an iSCSI array or NFS choosing "leave powered on" as your default isolation response might lead to a split-brain situation depending on the version of ESX used. The reason for this being that the disk lock times out if the iSCSI network is also unavailable. In this case the VM is being restarted on a different host while it is not being powered off on the original host. In a normal situation this should not lead to problems as the VM is restarted and the host on which it runs owns the lock on the VMX, but for some weird reason when disaster strikes you will not end up in a normal situation but you might end up in an exceptional situation. Do you want to take that risk?

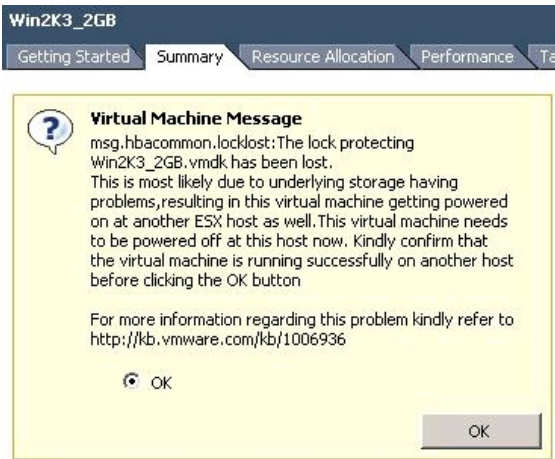
As of vSphere 4 Update 2 a new mechanism has been introduced which will recover VMs that are in a split brain situation. First let me explain what a split brain scenario is, let's start with describing the situation which is most commonly encountered:

- 4 Hosts
- iSCSI / NFS based storage
- Isolation response: leave powered on

When one of the hosts is completely isolated, including the Storage Network, the following will happen:

1. Host ESX001 is completely isolated including the storage network(remember iSCSI/NFS based storage!) but the VMs will not be powered off because the isolation response is set to "leave powered on".
2. After 15 seconds the remaining, non isolated, hosts will try to restart the VMs.
3. Because of the fact that the iSCSI/NFS network is also isolated the lock on the VMDK will time out and the remaining hosts will be able to boot up the VMs.
4. When ESX001 returns from isolation it will still have the VMX Processes running in memory and this is when you will see a "ping-pong" effect within vCenter, in other words VMs flipping back and forth between ESX001 and any of the other hosts.

As of version 4.0 Update 2 ESX(i) detects that the lock on the VMDK has been lost and issues a question which is automatically answered. The VM will be powered off to recover from the split-brain scenario and to avoid the ping-pong effect. The following screenshot shows the event that HA will generate for this auto-answer mechanism which is viewable within vCenter.



*Basic design principle 1: Isolation response should be chosen based on the version of ESX used. For pre-vSphere 4 Update 2 environment with iSCSI/NFS Storage I recommend to set the isolation response to "Power off" to avoid a possible split brain scenario. I also recommend to have a secondary service console running on the same vSwitch as the iSCSI network to detect an iSCSI outage and avoid false positives.*

*Basic design principle 2: Base your isolation response on your SLA. If your SLA dictates that hosts with degraded hardware should not be used, make sure to select shutdown or power off.*

## Isolation response gotchas

I thought this issue was something that was common knowledge but a recent blog [article](#) by Mke Laverick proved me wrong. I think I can safely assume that if Mke doesn't know this it's not common knowledge.

The default value for isolation/failure detection is 15 seconds. In other words the failed or isolated host will be declared dead by the other hosts in the HA cluster on the fifteenth second and a restart will be initiated by one of the primary hosts.

For now let's assume the isolation response is "power off". The "power off"(isolation response) will be initiated by the isolated host 1 second before the `das.failedetectiontime`. A "power off" will be initiated on the fourteenth second and a restart will be initiated on the fifteenth second.

Does this mean that you can end up with your VMs being down and HA not restarting them?

Yes, when the heartbeat returns between the 14th and 15th second the "power off" could already have been initiated. The restart however will not be initiated because the heartbeat indicates that the host is not isolated anymore.

How can you avoid this?

Pick "Leave VM powered on" as an isolation response. Increasing the `das.failedetectiontime` will also decrease the chances of running in to issues like these.

*Basic design principle: Increase "`das.failedetectiontime`" to 30 seconds (30000) to decrease the likely-hood of a false positive.*

## Admission control

This has always been a hot topic, HA and Slot sizes/Admission Control. One of the most extensive (Non-VMware) [articles](#) is by Chad Sakac aka Virtual Geek, but of course since then a couple of things has changed. Chad commented on this article and asked if I could address this topic, here you go Chad.

Lets start with the basics.

What's HAadmission control about? Why is it there? The "Availability Guide" states the following:

// *vCenter Server uses admission control to ensure that sufficient resources are available in a cluster to provide failover protection and to ensure that virtual machine resource reservations are respected.*

Admission Control ensures available capacity for HA initiated fail-overs by reserving capacity. Keep in mind that admission control calculates the capacity required for a fail-over based on available resources. In other words if a host is placed into maintenance mode, or disconnected for that matter, it is taken out of the equation. The same goes for DPM, if admission control is set to strict DPM in no way will violate availability constraints.

To calculate available resources and needed resources for a fail-over HA uses different concepts based on the chosen admission control policy.

Currently there are three admission control policies:

1. Host failures cluster tolerates
2. Percentage of cluster resources reserved as failover spare capacity
3. Specify a failover host

As stated each of these uses a different mechanism for reserving resources for a failover. Host failures uses a mechanism called "slots". Slots dictate how many VMs can be started up before vCenter starts yelling "Out Of Resources"! Normally each slot represents one VM.

Percentage based admission control uses a more flexible mechanism. It accumulates all reservations and subtracts it from the total amount of available resources while making sure the specified spare capacity is always available.

A failover host doesn't use any of those mechanisms, this host is dedicated for failover. It will not be used.

All three policies and concept will be explained in-depth below.

## Host failures

Now what happens if you set the number of allowed host failures to 1?

The host with the most slots will be taken out of the equation. (Slots are explained in more detail below) If you have 8 hosts with 90 slots in total but 7 hosts each have 10 slots and one host 20 this single host will not be taken into account. Worst case scenario! In other words the 7 hosts should be able to provide enough resources for the cluster when a failure of the "20 slot" host occurs.

And of course if you set it to 2 the next host that will be taken out of the equation is the host with the second most slots and so on.

One thing worth mentioning, as Chad stated with vCenter 2.5 the number of vCPUs for any given VM was also taken in to account. This led to a very conservative and restrictive admission control. This behavior has been modified with vCenter 2.5 U2, the amount of vCPUs is not taken into account.

*Basic design principle: Think about "maintenance mode". If a single host needs maintenance it will be taken out of the equation and this means you might not be able to boot up new VMs when admission control is set to strict.*

## What is a Slot?

// *A slot is a logical representation of the memory and CPU resources that satisfy the requirements for any powered-on virtual machine in the cluster.*

In other words a slot size is the worst case CPU and Memory reservation scenario in a cluster. This directly leads to the first "gotcha":

HA uses the highest CPU reservation of any given VM and the highest memory reservation of any given VM. If no reservations of higher than 256Mhz are set HA will use a default of 256Mhz for CPU and a default of 0MB+memory overhead for memory.

If VM1 has 2GHZ and 1024GB reserved and VM2 has 1GHZ and 2048GB reserved the slot size for memory will be 2048MB+memory overhead and the slot size for CPU will be 2GHZ.

*Basic design principle: Be really careful with reservations, if there's no need to have them on a per VM basis don't configure them.*

## How does HA calculate how many slots are available per host?

Of course we need to know what the slot size for memory and CPU is first. Then we divide the total available CPU resources of a host by the CPU slot size and the total available Memory Resources of a host by the memory slot size. This leaves us with a slot size for both memory and CPU. The most restrictive number is the amount of slots for this host. If you have 25 CPU slots but only 5 memory slots the amount of available slots for this host will be 5.

As you can see this can lead to very conservative consolidation ratios. With vSphere this is something that's

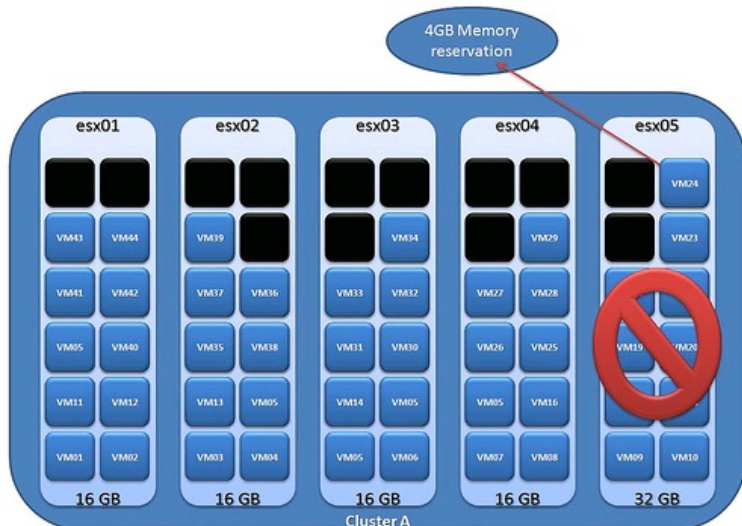


configurable. If you have just one VM with a really high reservation you can set the following advanced settings to lower the slot size being used during these calculations: `das.slotCpuInMHz` or `das.slotMemInMB`. To avoid not being able to power on the VM with high reservations these VM will take up multiple slots. Keep in mind that pre-vSphere 4.1 when you were low on resources this could mean that you were not able to power-on this high reservation VM as resources would be fragmented throughout the cluster instead of located on a single host.

As of vSphere 4.1 HA is closely integrated with DRS. When a failover occurs HA will first check if there are resources available on that host for the failover. If resources are not available HA will ask DRS to accommodate for these where possible. HA, as of 4.1, will be able to request a defragmentation of resources to accommodate for this VMs resource requirements. How cool is that?! One thing to note though is that HA will request it, but a guarantee can still not be given so you should be cautious when it comes to resource fragmentation.

The following is an example of where resource fragmentation could lead to issues:

If you need to use a high reservation for either CPU or Memory these options (`das.slotCpuInMHz` or `das.slotMemInMB`) could definitely be useful, there is however something that you need to know. Check this diagram and see if you spot the problem, the `das.slotMemInMB` has been set to 1024MB.



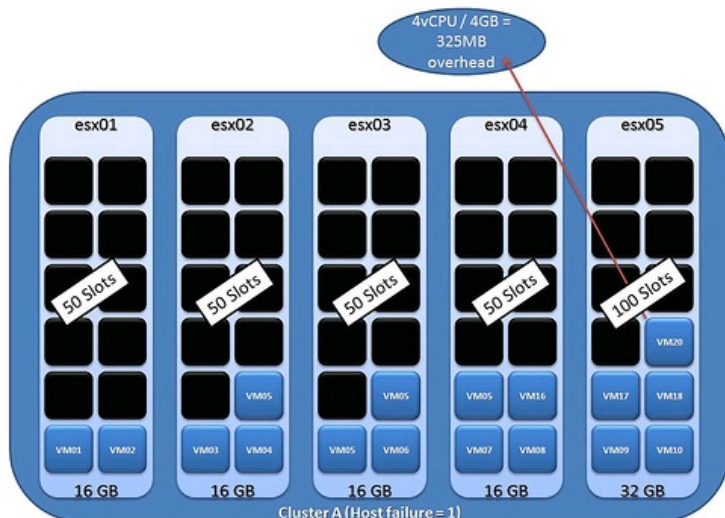
Notice that the memory slot size has been set to 1024MB. VM24 has a 4GB reservation set. Because of this VM24 spans 4 slots. As you might have noticed none of the hosts has 4 slots left. Although in total there are enough slots available; they are fragmented and HA might not be able to actually boot VM24. Keep in mind that admission control does not take fragmentation of slots into account when slot sizes are manually defined with advanced settings. It does count 4 slots for VM24, but it will not verify the amount of available slots per host. As explained, as of vSphere 4.1 it will request defragmentation, but as stated... it can not be guaranteed.

*Basic design principle: Avoid using advanced settings to decrease slot size as it might lead to more down time .*

Another issue that needs to be discussed is "Unbalanced clusters". Unbalanced would for instance be a cluster with 5 hosts of which one contains substantially more memory than the others. What would happen to the total amount of slots in a cluster of the following specs:

Five hosts, each host has 16GB of memory except for one host(esx5) which has recently been added and has 32GB of memory.

One of the VMs in this cluster has 4CPUs and 4GB of memory, because there are no reservations set the memory overhead of 325MB is being used to calculate the memory slot sizes. (It's more restrictive than the CPU slot size.)



This results in 50 slots for esx01, esx02, esx03 and esx04. However, esx05 will have 100 slots available. Although this sounds great admission control rules the host out with the most slots as it takes the worst case scenario into account. In other words; end result: 200 slot cluster. With 5 hosts of 16GB,  $(5 \times 50) - (1 \times 50)$ , the result would have been exactly the same. (Please keep in mind that this is just an example, this also goes for a CPU unbalanced cluster when CPU is most restrictive!)

*Basic design principle: Balance your clusters when using admission control and be conservative with reservations as it leads to decreased consolidation ratios.*

## Percentage of cluster resources reserved

Can I avoid large HA slot sizes due to reservations without resorting to advanced settings? Yes there is. The simplest way, without using advanced settings is selecting "Percentage of cluster resources reserved" as your admission control policy.

With vSphere VMware introduced a percentage next to an amount of host failures. The percentage avoids the slot size issue as it does not use slots for admission control. So what does it use?

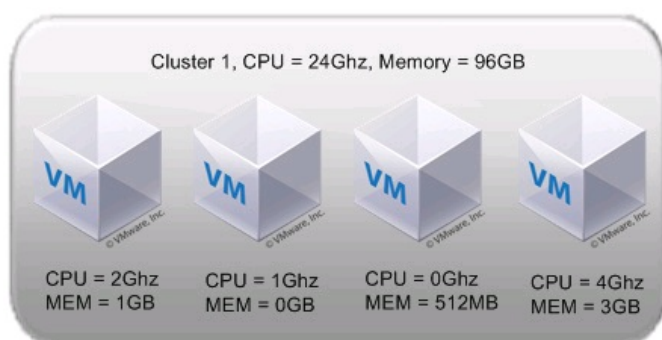
When you select a specific percentage that percentage of the total amount of resources will stay unused for HA purposes. First of all VMware HA will add up all available resources to see how much it has available. Then VMware HA will calculate how much resources are currently consumed by adding up all reservations of both memory and CPU for powered on virtual machines. For those virtual machines that do not have a reservation larger than 256Mhz a default of 256Mhz will be used for CPU and a default of 0MB+memory overhead will be used for Memory. (Amount of overhead per config type can be found on page 28 of the [resource management guide](#).)

In other words:

$((\text{total amount of available resources} - \text{total reserved VM resources}) / \text{total amount of available resources})$

Where total reserved VM resources include the default reservation of 256Mhz and the memory overhead of the VM.

Let's use a diagram to make it a bit more clear:



Total cluster resources are 24Ghz(CPU) and 96GB(MEM). This would lead to the following calculations:

$((24\text{Ghz} - (2\text{Gz} + 1\text{Ghz} + 256\text{Mhz} + 4\text{Ghz})) / 24\text{Ghz}) = 69\%$  available

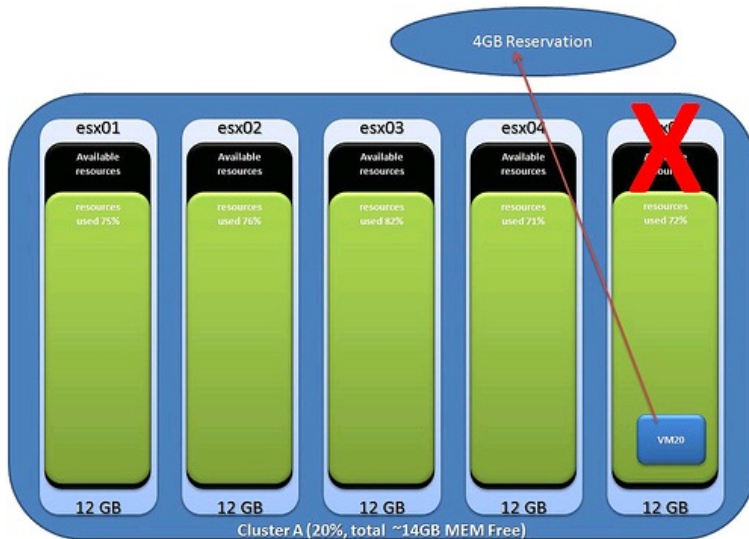
$((96\text{GB} - (1,1\text{GB} + 114\text{MB} + 626\text{MB} + 3,2\text{GB})) / 96\text{GB}) = 85\%$  available

As you can see the amount of memory differs from the diagram. Even if a reservation has been set the amount of memory overhead is added to the reservation. For both metrics HA admission control will constantly check if the policy has been violated or not. When one of either two thresholds are reached, memory or CPU, admission control will disallow powering on any additional virtual machines.

Please keep in mind that if you have an unbalanced cluster (host with different CPU or memory resources) your percentage is equal or preferably larger than the percentage of resources provided by the largest host. This way

you ensure that all virtual machines residing on this host can be restarted in case of a host failure. Another thing to keep in mind is as there are no slots which HA uses resources might be fragmented throughout the cluster. As explained earlier, HA will request DRS to defragment resource to cater for that specific VM, but it is not a guarantee. I recommend making sure you have at least a host with enough available capacity to boot the largest VM (reservation CPU/MEM). Also make sure you select the highest restart priority for this VM (of course depending on the SLA) to ensure it will be able to boot.)

I created a diagram which makes it more obvious I think. So you have 5 hosts, each with roughly 76% memory usage. A host fails and all VMs will need to failover. One of those VMs has a 4GB memory reservation, as you can imagine failing over this particular VM will be difficult due to the fact that none of the hosts has enough memory available to guarantee it. Although HA will request DRS to free up resources it is not guaranteed DRS can actually do this.



*Basic design principle: Do the math, verify that a single host has enough resources to boot your largest VM. Also take restart priority into account for this/these VM(s).*

#### Specify a failover host

With the Specify a Failover Host admission control policy, when a host fails, HA will attempt to restart all virtual machines on the designated failover host. The designated failover host is essentially a "hot standby". In other words DRS will not migrate VMs to this host when resources are scarce or the cluster is imbalanced.

## My Admission Control Policy Recommendation

It depends. Yes I know, that is the obvious answer but it actually does. There are three options and each have it's own advantages and disadvantages. Here you go:

- Amount of host failures
  - Pros:
    - Fully automated, when a host is added to a cluster HA calculates how many slots are available.
    - Ensures fail-over by calculating slot sizes.
  - Cons:
    - Can be very conservative and inflexible when reservations are used as the largest reservation dictates slot sizes.
    - Unbalanced clusters leads to waste of resources.
- Percentage reserved
  - Pros:
    - Flexible as it considers actual reservation per VM.
    - Cluster dynamically adjusts number of host failure capacity when resources are added.
  - Cons:
    - Manual calculations need to be done when adding additional hosts in a cluster and amount of host failures need to remain unchanged.
    - Unbalanced clusters can be a problem when chosen percentage is too low.
- Designated failover host
  - Pros:
    - What you see is what you get.
    - High resource utilization as dedicated fail-over host is unused.
  - Cons:
    - What you see is what you get.
    - Maximum of one fail-over host.
    - Dedicated fail-over host not utilized during normal operations.



Basic design principle: Do the math, and take customer requirements into account. if you need flexibility a "Percentage" is the way to go.

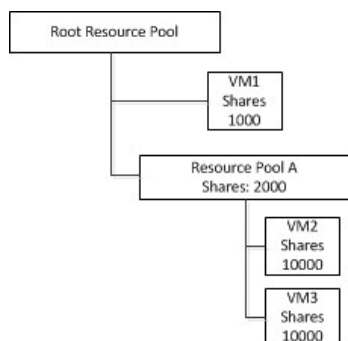
## Flattening Shares

Prior to vSphere 4.1, an HA failed over virtual machine could be granted more resource shares then it should causing resource starvation until DRS balanced the load. As of vSphere 4.1 HA calculates normalized shares for a virtual machine when it is powered on after an isolation event!

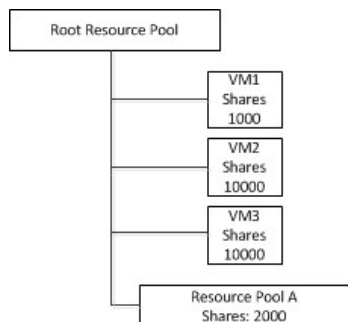
Pre-vSphere 4.1 an issue could arise when shares had been set custom on a virtual machine. When HA fails over a virtual machine it will power-on the virtual machine in the Root Resource Pool. However, the virtual machine's shares were scaled for its appropriate place in the resource pool hierarchy, not for the Root Resource Pool. This could cause the virtual machine to receive either too many or too few resources relative to its entitlement.

Scenario where and when this can occur would be the following:

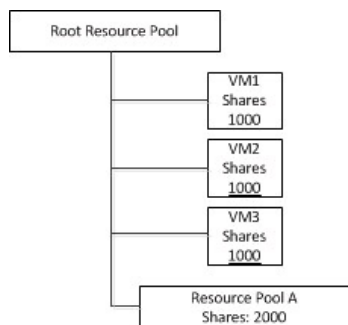
VM1 has a 1000 shares and Resource Pool A has 2000 shares. However Resource Pool A has 2 VMs and both will have 50% of those "2000" shares.



When the host would fail both VM2 and VM3 will end up on the same level as VM1. However as a custom shares value of 10000 was specified on both VM2 and VM3 they will completely blow away VM1 in times of contention. This is depicted in the following diagram:



This situation would persist until the next invocation of DRS would re-parent the virtual machine to its original Resource Pool. To address this issue as of vSphere 4.1 DRS will flatten the virtual machine's shares and limits before fail-over. This flattening process ensures that the VM will get the resources it would have received if it would have been failed over to the correct Resource Pool. This scenario is depicted in the following diagram. Note that both VM2 and VM3 are placed under the Root Resource Pool with a shares value of 1000.



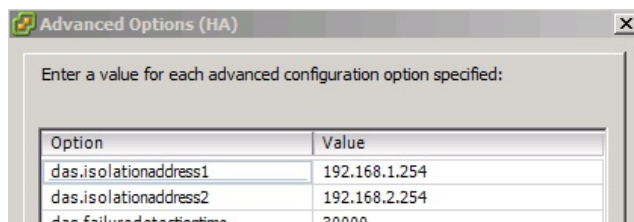
Of course when DRS is invoked both VM2 and VM3 will be re-parented under Resource Pool A and will receive the amount of shares they had originally assigned again. I hope this makes it a bit more clear what this "flattened shares" mechanism actually does.

## Advanced Settings

VMware HA is probably the feature with the most advanced settings. Although many of them are rarely used some of them are needed in specific situations or included in best practices documents. The most used and valuable advanced settings are described below:

das.failedetectiontime – Amount of milliseconds, timeout time, for isolation response action (with a default of 15000 milliseconds). It's a best practice to increase the value to 60000 when an active/standby Service Console setup is used. For a host with two Service Consoles and a secondary isolation address it's a best practice to increase it to at least 20000. I would recommend to always increase it to at least 30000

das.isolationaddress[x] – IP address the ESX hosts uses to check on isolation when no heartbeats are received, where [x] = 1-10. (see screenshot below for an example) VMware HA will use the default gateway as an isolation address and the provided value as an additional checkpoint. I recommend to add an isolation address when a secondary service console is being used for redundancy purposes.



Option	Value
das.isolationaddress1	192.168.1.254
das.isolationaddress2	192.168.2.254
das.failedetectiontime	30000

das.usedefaultisolationaddress – Value can be true or false and needs to be set in case the default gateway, which is the default isolation address, should not or cannot be used for this purpose. In other words, if the default gateway is a non-pingable address set the “das.isolationaddress” to a pingable address and disable the usage of the default gateway by setting this to “false”.

das.allowVmotionNetworks – Allows a NIC that is used for Vmotion networks to be considered for VMware HA heartbeat usage. This permits a host (ESXi only) to have only one NIC configured for management and Vmotion combined.

das.allowNetwork[x] – Enables the use of port group names to control the networks used for VMware HA, where [x] = 0 – ?. You can set the value to be [Service Console 2] or [Management Network] to use (only) the networks associated with those port group names in the networking configuration. These networks need to be compatible for HA to configure successfully.

das.maxvmrestartcount – Configure the maximum amount of retries for a restart of a virtual machine. Select the value “0” for no restarts or -1 for indefinite. The default is 5 and has been chosen to avoid the issues described in [KB 1009625](#).

das.bypassNetCompatCheck – Disable the “compatible network” check for HA that was introduced with Update 2. Default value is “false”, setting it to “true” disables the check. This setting can be useful when nodes in a cluster are not in the same subnet.

das.ignoreRedundantNetWarning – Remove the error icon/message from your vCenter when you don't have a redundant Service Console connection. Default value is “false”, setting it to “true” will disable the warning.

*Basic design principle: Avoid using advanced settings as much as possible as it leads to increased complexity.*

## 43 Responses to “HA Deepdive”



**sconyard** says:

Monday, July 27, 2009 at 16:05

So in essence; the 'Leave powered on' catches two of the three HA outage types, false positives covered, host down covered, but complete network failure, when the COS and the VM networks go offline, Would leave VMs running without networking and require manual intervention.



**Mike Laverick** says:

Monday, July 27, 2009 at 17:03

Duncan,

Thanks for this. About this 12/15 second thang. What benefit to VM or generally does this offer. Why not have a timeout value of 15 seconds, and then this 13/14th second anomaly would “go away”. I'm struggling to understand why HA uses this 3 second offset?



**Chad Sakac** says:

Monday, July 27, 2009 at 17:08

thanks Duncan – was planning on doing a post on this topic (as my old HA articles are also popular, but out of date. Would like to start directing them to more current docs...

I think you need to include a short discussion on slot size calculations – as this is part of HA, and is one of those “unknown important internals” (that also happens to keep changing).



**Duncan says:**

Tuesday, July 28, 2009 at 11:29

@sconyard: agree, it does catch these

@mike laverick: I think it's because VMs need to be powered off before they can be restarted so there's always the possibility that the heartbeat returns before the restart is initiated. A better solution might be: declare dead, shutdown, if heartbeat returns ignore, restart. if host/heartbeat is consistent for 5 minutes add in to the HAloop again.

@chad: good idea, when I can find the time I will add it for sure. or if you've got something laying around let me know...



**Serge Meeuwssen says:**

Saturday, August 1, 2009 at 14:04

@sconyard

Not sure if I completely understood what you meant, but in theory your third scenario is actually also covered by design as HA does not only look at a lock file, there is also a heartbeat written to the SAN. When the host dies it won't update this heartbeat counter thus surviving hosts have a way of aging the lock. Of course in the event of a complete network failure the restarted VMs will probably be inaccessible as it is very likely the surviving hosts will also be impacted...



**Duncan says:**

Saturday, August 1, 2009 at 19:43

Ha does not send a heartbeat to the SAN that's default ESX behavior to prevent the lock from staling and prevent users from starting the VM twice.



**Serge Meeuwssen says:**

Monday, August 3, 2009 at 13:55

@Duncan

Sorry for muddying the waters 😊 you're right of course this 'heartbeat' to SAN behavior is indeed default ESX behavior managed by the Distributed Lock Manager and not part of HA



**Halibut says:**

Wednesday, August 12, 2009 at 15:55

Intriguing article, it is always helpful to deep dive into an aspect of the software and understand some of the advanced settings. I would also like to know where HA files reside with the service console to perform HA troubleshooting. I have had a few sites that have a recurring HAerror, sometimes disabling and re enabling the client sometimes works but not always.



**dconvery says:**

Saturday, September 19, 2009 at 20:33

Duncan –

I'm just trying to clarify slot size calculations and how to roll it up to figuring out the number of requires servers in an HA cluster.

Lets say the largest VM is 4CPU and 16GB RAM (with a 16GB reservation). The ESX servers are all 2 socket, quad core 3GHz with 32GB RAM

With overhead (about 650), my RAM slot size would end up being around 17GB., which gives me less than two slots per ESX server. Is this correct?

Now, if I have a VM with only 1 CPU and 2GB RAM, it will still take up a slot. If I do not change the default slot size settings, the slot is roughly 75% wasted. Is this correct?

In this scenario, if I do not "tweak" the slot sizes, do I only get one VM per node since it works out to about 1.7 slots per node?

Dave



**Johnny Z says:**

Sunday, September 20, 2009 at 05:03

1. You can find out the primaries by looking at  
`/var/log/vmware/aam_config_util_listprimaries.log`
2. One of the primaries will be the rule cluster manager which hold all the rules for this HAcluster. You can find this out by "grep -i submitted /var/log/vmware/aam/vmware\_server\_name.log (You need replace server\_name with your own server's name)
3. If you want to know the step by step how HA add a host take a look at  
`/var/log/vmware/aam/aam_config_util_addnode.log`  
This log will show you every step how HA add a host into the HAcluster, and if there is a problem it will also tell you where the problem is. In many situation you can not do much but reconfigure it, but it will give you an idea on which step went wrong



**Johnny Z** says:

Sunday, September 20, 2009 at 05:17

One more thing, when the heartbeat lost on one of the HA host, this could be a real network problem/hardware failure or just HAagent stopped for some reason. It's not a good idea to power on all the VM when you lost heartbeat, so what the other hosts will do is "PING" the problem host to confirm. If the host reply to the ping, it will mark as ALIVE and the other hosts does nothing. If it didn't then it will mark as DEAD, and the other hosts will ping the gateway to make sure themselves are good then power on the VMs. You can see that from less vmware\_server\_name.log | grep -i "Ping Node results:"



**Duncan** says:

Sunday, September 20, 2009 at 08:51

Keep in mind that unfortunately the log file does not always show the most current state that's why I listed the other option which uses the CLI and should always be the most current state.

Thanks for your comments.



**Johnny Z** says:

Sunday, September 20, 2009 at 14:41

Hi Duncan,

HA will keep 10 aam\_config\_util\_listprimaries.log files, and the latest one would reflect the last change.  
For the Cli, you do need to export:

```
export FT_DIR=/opt/vmware/aam/  
export FT_DOMAIN=vmware
```

or it will not work



**Jason Boche** says:

Thursday, October 8, 2009 at 22:28

vSphere:

The memory slot size is determined by the largest sum of a VM memory reservation plus the memory overhead of the VM

By default VMs have no memory reservations.

In the case of a cluster full of VMs with no memory reservation, the memory slot size will actually be the largest memory overhead of a VM (ie. 88MB).

Jas



**Mike Laverick** says:

Monday, October 26, 2009 at 22:19

I'm looking at this whole slot thing all over again. It's horrible complicated isn't? I wonder how many customers look at this – and look for a different way of representing free capacity for fail-over such as the new percentage value...



**Duncan Epping** says:

Monday, November 9, 2009 at 17:45

agreed Mike, with the Percentage option it does make it a bit less complicating.



**Edvard Hofmann** says:

Monday, December 21, 2009 at 22:26

Can you further explain the calculation giving the memory overhead of 325 MB given 4 vCpu / 4 GB in section "Slot size/Admission Control".



**Duncan Epping** says:

Monday, December 21, 2009 at 22:33

There's not much to explain... just check page 28 of the resource management guide: [http://www.vmware.com/pdf/vsphere4/r40\\_u1/vsp\\_40\\_u1\\_resource\\_mgmt.pdf](http://www.vmware.com/pdf/vsphere4/r40_u1/vsp_40_u1_resource_mgmt.pdf)



**Edvard Hofmann** says:

Monday, December 21, 2009 at 23:00

Hi again,

There it is, I review the resource mgmt guide for V3 by mistake: [http://www.vmware.com/pdf/vi3\\_301\\_201\\_resource\\_mgmt.pdf](http://www.vmware.com/pdf/vi3_301_201_resource_mgmt.pdf).

Thanks.

Edvard



**Menno** says:

Monday, January 18, 2010 at 13:00

Thanks for sharing all this information on HA

I'm wondering if there is a way to put HA in a testing mode.

In our current setup we unfortunately sometimes have network hick-ups. This made HA make some unwanted decisions. So I've stopped HA. Now to cope with these network issues, I've adjusted some advanced HA settings, mainly to make HA a bit more flexible.

Do you know if it's possible to have HA in a sort of simulation mode and just report on what it wants to do, instead of actually doing it? So I can study its behaviour for a certain period and then later decide to go live with it.



**Chuck** says:

Wednesday, January 20, 2010 at 23:32

At some point could you address how the surviving HA nodes determine which VMs were affected by the host failure, whether the method changes if vCenter is unavailable, and how/where that state is maintained and updated?

We recently had an issue where we moved most of our VMs from one array to another and experienced a host failure a few days later. During the HA failover the HA agent on multiple servers attempted to register several VMs using their old datastore locations and subsequently failed to bring them back. Fortunately the vCenter VM did restart successfully and we were able to manually power on the remaining VMs.



**Duncan Epping** says:

Saturday, January 23, 2010 at 08:26

State is registered in a "database" and part of the node state and replicated amongst each other. vCenter isn't used during the failover at all. HA isn't dependent on vCenter in any way when it is up and running.

What happened in your specific case I really don't know...



**Adam Baum** says:

Monday, January 25, 2010 at 16:54

What types of items can cause isolation responses other than an actual NIC and/or Network failure? I have a single service console per host, each with two pNICs. pNICs are separated onto two physical switches (same VLAN). I've set the options mentioned above and provided numerous isolation addresses to ping physical devices on each switch. With all these tuning parameters, I still encountered an isolation event. Nothing in the alarm logs says what happened as far as I can tell, just that I had one.

One more bit of info... One of the physical switches did have a problem. We don't know what yet since it appears to still have the problem, but every server is running fine. So... why didn't the second pNIC kick in? I



have both NICs in an active/active config. Should they have been in an active/standby config?

adam



**Doug says:**

Wednesday, February 3, 2010 at 18:27

FWIW, I was looking into reporting which nodes hold the Primary HA roles in my environment and had a problem doing it without logging into the console of an ESX/ESXi host. You can get a list of the HA Primaries for a given cluster fairly quickly using one line in the PowerCLI:

```
((Get-View  
YourClusterName).id).RetrieveDasAdvancedRuntimeInfo().DasHostInfo.PrimaryHosts
```



**DeadParrot says:**

Thursday, February 4, 2010 at 17:10

Regarding 'das.ignoreRedundantNetWarning' advanced setting....

In ESXi this would be redundancy on the Management network?

Simon



**Dan says:**

Wednesday, February 17, 2010 at 08:59

Is there a way to keep a specific VM on a specific host once HA have been implemented. We used to do this by CPU affinity, but this is not available?

The need to do this arises because we have hosts in two buildings, in each a VM monitors an UPS and runs a script to shutdown the hosts in that building in case of power failure.

Dan



**Michael Poore says:**

Monday, February 22, 2010 at 14:53

Just thought I'd mention that Alan Renouf has written a short PowerCLI script that displays information about slot sizes. Apologies if this is already mentioned somewhere here but I didn't spot it. Thought it might be useful.

<http://www.virtu-al.net/2009/10/06/ha-slot-size-information/>



**Phil Maynard says:**

Monday, March 8, 2010 at 18:31

For those of you that would like to add a layer of application awareness to VMware HA, you might like to check out vAppHA: –

<http://www.neverfailgroup.com/virtualization/vapphatrial.html>



**Blackdevile says:**

Thursday, April 22, 2010 at 12:18

Hello,

thanks for this great explaining of HA

I just started in my company to work with ESX as the new Administrator and this one helps me to understand much better what happens inside of my servers.

Greetz

chris



**Robert says:**

Thursday, April 22, 2010 at 22:24

If you have multiple Isolation IP addresses/networks, when HA starts to ping the isolation IP address, does it ping all of them (default and additional) simultaneously or in series?



**ramdraft** says:

Friday, April 23, 2010 at 20:58

Thanks for sharing your knowledge with us. However we couldn't find an answer to the question: How do we configure HA to prepare for a split-brain situation given the following facts:

- The DRS/HA cluster is stretched across two physical sites (10km distance).
- The DRS/HA cluster contains 8 ESX HA-nodes (4 on each site) running on synchronously mirrored storages and both storage servers are write enabled on the very same VMFS-LUN.
- only one single HA gateway is configured to an IP on site A
- The split-brain is a sudden event, all management and FC-lines are chopped at the very same moment.

The problem is: Since all HA-nodes at site B can still heartbeat each other, they don't power off their VMs. They consider themselves to be the major surviving part, even though the gateway is not reachable anymore. Our goal: Finding an HA advanced option to command the VM to power off because the HA gateway is unreachable.

Thank you very much in advance for any suggestions.



**duncan** says:

Saturday, April 24, 2010 at 07:56

Sorry, that is currently not possible... there is no such an advanced option. it will always check for incoming heartbeats and having it split across two sites means there is always traffic coming in from a single site.



**Tim Fort** says:

Monday, May 3, 2010 at 16:48

Doug,

Thank you for the vCLI HAnode command... that has come in quite handy.

Duncan, thanks for everything as always!



**ramdraft** says:

Friday, May 7, 2010 at 22:21

Thanks for confirming the split issue from HA perspective. We've come up with a solution. Basically we trigger a switch IOS to release all management ports of the ESX on site A to make them "feel isolated". That's way quicker than 15 seconds and allows to have a stretched cluster with no headache. Cheers & keep it up 😊



**Tom** says:

Thursday, June 17, 2010 at 18:21

"Basic design principle: For iSCSI I recommend to set the isolation response to "Power off" to avoid a possible split brain scenario. I also recommend to have a secondary service console running on the same vSwitch as the iSCSI network to detect an iSCSI outage and avoid false positives."

I remember reading that vSphere 4.0 Update 1 or 2 fixes the split brain issue.

Hence it's unclear to me whether your suggestion should be followed.

Could you perhaps find time to quickly diagram what/where your suggestion of a secondary Service Console fits into the networking setup??

My understanding is that the secondary Service Console would be in the same vSwitch as the backend iSCSI vmKernels, one SC per vmKernel, each with its own IP address.

Any illumination you could shed would be most helpful.

Thank you, Tom



**Nicolas** says:

Wednesday, June 23, 2010 at 10:33

Hi Duncan,

How long does it take for a .vmdk lock to be released when the esx loses its connection to the SAN?

is it the same time in FC, iscsi and NFS?

and is it possible to modify it?

Thank you and congrats for your work on yellow-bricks !



**AIIBlack** says:

Thursday, July 22, 2010 at 02:23

Great article. I have a few questions about the slot size. I have no reservations in my cluster. The biggest VM has 8gb of RAM and 2vCPU. According to the Resource Management guide that is an overhead of 331 MB. However in vCenter my HA run time info says my memory slot size is 775Mb?

Why would this be?

Also, am I correct in the assumption that, if my slot size is i.e 200Mb I would require 10 slots to boot a 2Gb VM?



**Duncan Epping** says:

Thursday, July 22, 2010 at 13:01

@AIIBlack: That is weird... Can't explain that.

If you set your slot size to 200MB manually and you have a VM with a 2GB reservation you would need "10 slots" indeed. But this hardly ever happens I guess. Keep in mind that it is all about setting resources aside for a possible failover.



**Adam Wallport** says:

Thursday, July 22, 2010 at 18:28

Hi Duncan, I'm trying to understand the best strategy here given that we really want HA to work without risk of VMs not starting. Slots seem very restrictive given that we have things like Nexus 1000v insisting on 1500Mhz and 2GB and if we use advanced settings to trim the slot size we still need multiple slots for big VMs with the risk of fragmentation (is there a way to force big VMs to start first btw as this would seem to avoid this risk?). I have read somewhere of someone using resource pools to assist but can't say I understand how resource pools interact with HA slots.

The other factor is using percentage based admission control but this sounds like it relies on manual calculation in what may be a dynamic environment and although (as I understand it) it does away with slots, it sounds like it still has risks associated of not being able to start individual VMs.

Please could you help me understand which approach is best given that we want VMs to start after a node loss and don't want too much manual intervention?

TIA



**Duncan Epping** says:

Friday, July 23, 2010 at 12:37

The risk is not as big as it seems. I still need to update the above post but with 4.1 fragmented resources are defragmented by HA when needed. although not a guarantee it is as much risk as when doing a fixed manual slot size in my opinion.



**marcus** says:

Thursday, July 29, 2010 at 15:58

Hi Duncan,

With the recent release of ESX 4.1 does the following design principle still hold true?

"Basic design principle: In blade environments, divide hosts over all blade chassis and never exceed four hosts per chassis to avoid having all primary nodes in a single chassis."

Does ESX 4.1 now support more than 5 primary nodes in HA?

With ESX 4.1, in a VMware n+2 cluster with 8 VMware ESX hosts... if say two VMware ESX hosts fail simultaneously, will VMs from failed VMware ESX host 1 get started first, then only when complete, will VMs from failed VMware ESX host 2 start? Imagine losing 4 VMware ESX hosts in a 16 host cluster spread across 4 blade chassis... would take a lot longer to restart VMs on a failed host by failed hosts basis... rather than being able to restart VMs from all 4 failed VMware ESX hosts simultaneously on the remaining 12 hosts ...

Regards,  
Marcus



**Justin Guidroz** says:

Thursday, July 29, 2010 at 16:57

@Adam Wallport

The easy solution with the Cisco Nexus 1000V is to remove the per VM resources for memory and CPU and create a resource pool for the VMs with these same settings. Slot sizes for your cluster will return to normal.



**Jose A. Hernandez** says:

Wednesday, August 11, 2010 at 01:47

Hi Duncan,

I'm still wrapping my head around this. It seems this goes out the window when the cluster is made up of only 2 hosts. Can you confirm this? I have a 2 host test cluster with 16GB and 4 Cores per host for a total of 32GB and 8 cores. There are 4 VMs on it:

2 VMs are 1vCPU and 2GB

1 VM is 2vCPU and 2GB

last VM is 1vCPU and 4GB

Only the 2 VMs are powered on (the 1vCPU/4GB and the 2vCPU/2GB VMs). These 2 VMs are split between hosts. My problem is that I can not complete a host remediation via VMware Update Utility because when it tries to put one of the 2 hosts in maintenance mode, DRS barks stating that there are insufficient resources to satisfy HA. Admission Control is enabled and my HA 'Admission Control Policy' is set to 'Host Failures cluster tolerates' = 1.

VMware support just told me that because this is a 2 node cluster, user initiated request to put a host on maintenance mode will fail with this setting, regardless if the remaining host had enough resources to run the VM that need to migrate. In the case of an actual failure, they say, the VMs will migrate.

They stated it should work if I use the percentage option instead but that doesn't seem to work either. I set the percentage at 25%. The cluster showed a CPU Fail-over Capacity of 97% and Memory Fail-over capacity of 98%. Yet when I tried to put one host into maintenance mode it failed again with the same error about insufficient resources. This seems wrong to me. The only way I can update the hosts is by disabling the HA Admission Control Policy.

We have 14 remote sites with 2 hosts ESX clusters on them so this has some annoying repercussions in my environment.

## Leave a Reply

Name (required)

Mail (will not be published) (required)

Website

Submit Comment

Copyright © Yellow Bricks - Building blocks for virtualization...

Powered by WordPress | Modern Style theme by FlexiThemes