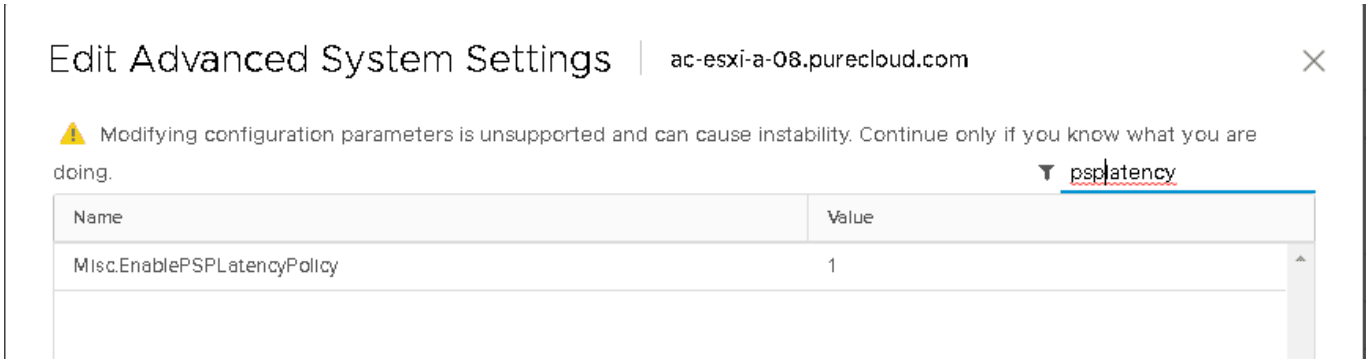
Latency Round Robin PSP in ESXi 6.7 Update 1



This is my first (but certainly not last post) on the new path selection policy option in vSphere 6.7 Update 1. In reality, this option was introduced in the initial release of 6.7, but it was not officially supported until update 1.

So what is it? Well first off, see the official words from my colleague Jason Massae at VMware here

Why was this PSP option introduced? Well the most common path selection policy is the NMP Round Robin. This is VMware's built-in path selection policy for arrays that offer multiple paths. Round Robin was a great way to leverage the full performance of your array by actively using all of the paths simultaneously. Well…almost simultaneously.

The default configuration of RR was to switch logical paths every 1,000 I/Os for a given device. So for a given device it would use one path for 1,000 I/Os, then the 1,001st I/O would go down a different path. And so on.

The other option was to change paths after a certain amount of throughput, but frankly, very few went down that route.

A popular option for tuning RR, was to increase the path switching frequency. By changing what was called the I/O Operations Limit (sometimes called the IOPS value) you could realize a few additional benefits. The reason we (Pure) recommended the change down to switching paths after every single I/O was for two many reasons:

1. Path failover time. When a path failed due to some physical failure along that path (switch, HBA, port, cable, etc) ESXi would fail that path much more quickly leading to less disruption in performance during the failure
2. I/O balance. When this was set low, the I/O balance on the array (and from a host) was usually almost perfectly balanced across the paths. This made it much easier to identify when something was configured wrong (not fully zoned etc).

A third argument was performance, but frankly there isn't a lot of strong evidence for that. But nonetheless

the other benefits caused this change to generally be recommended. Almost every storage vendor who offers active/active paths made this recommendation too.

With that all being said, it still wasnt quite good enough. Not all paths are created equal, or more importantly, stay equal.

Failures, congestion, degradation, etc. could cause one or more of the available paths to not go offline, but behave erratically. In other words, I/Os sent down the misbehaving paths experienced worse performance than I/Os sent down healthy paths. Generally, this meant the latency down the bad path was bad, the latency on the good paths was, well, good.

The issue here is that round robin only stopped using a path when it was dead. If the latency on one path was 100 ms and .5 ms on the other, it would see and use each path equally. As long as it was "online" it was a valid and active path. You can see how this could be a problem. In these situations, somewhat ironically, if ESXi used *less* paths, the performance would improve. By skipping the unhealthy paths.

With the introduction of all-flash-arrays, latency become more and more a centerpiece of the conversation. With the burgeoning support for NVMe-oF, even lower latency is possible. VMware saw this, and started working on a new policy. So they introduced a new dynamic and adaptive latency-based policy.

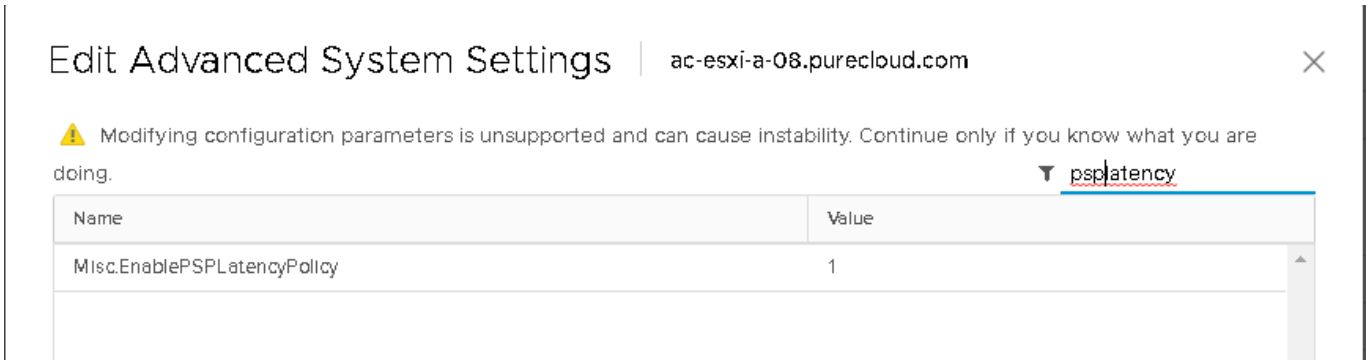
In ESXi 6.7 U1, there is a new latency-based option for the round robin PSP (path selection policy). When enabled, ESXi will sample the paths every 3 minutes with 16 I/Os. It will then calculate the average latency for those I/Os and decide (in comparison to the other paths) whether or not to use that path. If it is deemed too unhealthy, it will be excluded until the next sampling period begins in 3 minutes where it will be re-evaluated.

Jason's post talks about using the CLI and host profiles to set this policy, so I will not cover it here. What I have not seen yet is two things, setting it with PowerCLI and also setting it to be a default using SATP rules.

I plan on writing a more in-depth post with some testing scenarios and results in the near future, but at least in this post let me make a few things clear:

- **Does Pure Storage recommend this as a best practices moving forward?** Not at this time. Not because it doesn't work (it does and we have seen great results), but because we need to put a few more miles on it. We will start using it in our standard QA environments etc. When we are ready for it to be our best practice–we will work with VMware to make it our default in ESXi.
- **Does Pure Storage support it?** Yes. VMware supports it and so do we. Our recommendation is the same as any new feature–test it first before using in prod.
- **What are the recommended settings for the latency PSP?** At this point, the default. I doubt we will make recommendations otherwise. You can change the sample size and interval, but we have yet to see a reason to deviate from the default and it is unlikely that will change. If it does, I will certainly post about it. More testing needs to be done on our side to understand these settings more fully.

One thing I will note, is I have already seen some erroneous information about enabling this feature. There is an advanced setting called **Misc.EnablePSPLatencyPolicy** that must be set to 1. This was set to 0 in 6.7 GA, but for ESXi 6.7 Update 1–this is enabled by default in update 1 and can also be disabled/enabled in the GUI:



I have tested upgrading from 6.5 and 6.7 GA to 6.7 update 1 and this setting gets set to 1 upon upgrade.

This only needs to be set in the first 6.7 release–but since it is not officially supported, I would just upgrade instead.

UPDATE: See a deeper post on testing/use cases here:

Latency-based PSP in ESXi 6.7 Update 1: A test drive

Configuring the Latency Policy as a SATP Rule in ESXCLI

If you want to change your test/dev environment to this policy, the best way is with a SATP (storage array type plugin) policy. This allows you to say (for instance) “hey for all Pure Storage FlashArray devices, use the round robin policy with the latency path selection option”. This way only Pure Storage FlashArray devices get it–other arrays that might not want it won’t. Furthermore, all new Pure Storage devices (volumes, datastores, RDMs, whatever) will automatically get that policy.

Definitely a handy feature. Set it and forget it.

In order to set it, SSH into your ESXi host and run the following:
[crayon-6515c48354424413509144/]
This will make sure that all Pure Storage FlashArray storage volumes provisioned to that host from that point on will be configured with the latency policy.

I add that SATP rule then created a new datastore and you can see with the following command that it is indeed configured with that policy:
[crayon-6515c48354434373065888/]

```
naa.624a93701037b35fd0ef40a50005e0c7
   Device Display Name: PURE Fibre Channel Disk (naa.624a93701037b35fd0ef40a50005e0c7)
   Storage Array Type: VMW_SATP_ALUA
   Storage Array Type Device Config: {implicit_support=on; explicit_support=off; explicit_allow=on; alua_followover=on; actio
   Path Selection Policy: VMW_PSP_RR
   Path Selection Policy Device Config: {policy=latency,latencyEvalTime=180000,samplingCycles=16,curSamplingCycle=4,useANO=0;
   Path Selection Policy Device Custom Config:
   Working Paths: vmhba2:C0:T7:L249, vmhba2:C0:T6:L249, vmhba2:C0:T9:L249, vmhba2:C0:T5:L249, vmhba2:C0:T8:L249, vmhba2:C0:T4
T5:L249
```

Configuring the Latency Policy as a SATP Rule in PowerCLI

The above is nice, but it is just for a single host. Using PowerCLI is a great way to set it for many hosts at once. Using the get-esxcli cmdlet, you can easily set this on an entire cluster or datacenter or even vCenter if you so choose.

Connect your vCenter and for each host, run the following:

[crayon-6515c48354437965608218/]
So put that in a for loop or whatever. Obviously that is for Pure, so if you have a different vendor, change the model and vendor name.

Stay tuned for more info!