PURESTORAGE®

BROADCOM® EMULEX® Fibre Channel HBAs

vmware®

TECHNICAL WHITE PAPER

# Adaptive vVol Performance Management

Experience double the performance and half the latency for vVols with Emulex's latest FC HBA drivers.

# Contents

## Introduction

While using virtual volumes (vVols) in your vSphere environment, you may have noticed that the queue depth was set too low, resulting in you needing to adjust the protocol endpoint (PE) queue depth manually on the ESXi hosts. With Emulex's current Fibre Channel (FC) host bus adapter (HBA) drivers, the PE queue depth is automatically adjusted to 254. This white paper discusses the performance improvements associated with this change and other considerations.

## Why Does this Change Matter?

Consolidating virtual machine file system (VMFS) datastores into a smaller number of VVol storage containers reduces complexity and makes VMware storage management simpler. Additionally, vVols provide a more sophisticated way of managing fairness through QoS-based management at a VM granular level. However, VMware administrators may have observed the default queue depth on the PE set to 128 or lower. This can, however, potentially cause performance bottlenecks due to the increased density of VMs on the same queue with vVols.

This can be lower if the HBA's default max queue depth limit is below 128 or it is manually set below 128 by a VMware administrator. With the updated Emulex driver, the default queue depth limit is 254 for each PE.

## Testing Environment

For the purposes of this white paper, the test environment comprised of the following:

- One Pure Storage® FlashArray™ x50r3 on Purity 6.1.8
- One vCenter 7 U2 server on build 7.0.2.00200
- One ESXi 7 U2 host on build 17630552
- One Emulex LightPulse LPe36000 PCIe Fibre Channel HBA on driver version 12.8.614.3-1OEM.700.1.0.15843807_18994299
- Two standard Fibre Channel switches; all host and array ports connected at 16Gb/s
- Two connected 16Gb/s ports per FlashArray controller, totaling four connected 16Gb/s Fibre Channel ports
- Two connected 16Gb/s ports per Emulex HBA, totaling two 16Gb/s ports on the ESXi host
- One Windows Server 2019 VM with 16 CPUs and 32GB of memory running VDbench to generate workload

The minimum requirements of the automatic PE queue depth limit configuration are:

- Emulex driver version 12.8.534.0
- ESXi 7
- Emulex HBA hardware model listed in the VMware compatibility guide reference link

Let's get a clear understanding of where we are from a device perspective before we get into the test results.

First, let's look at the queue depth limits for our Emulex-backed devices. To get the device ID for our vVol container and VMFS datastore, we need to log in to the vCenter GUI.

For our VMFS datastore:

1. Click on the **Storage** view
2. Select our VMFS datastore
3. Click **Configure**
4. Click **Device Backing**

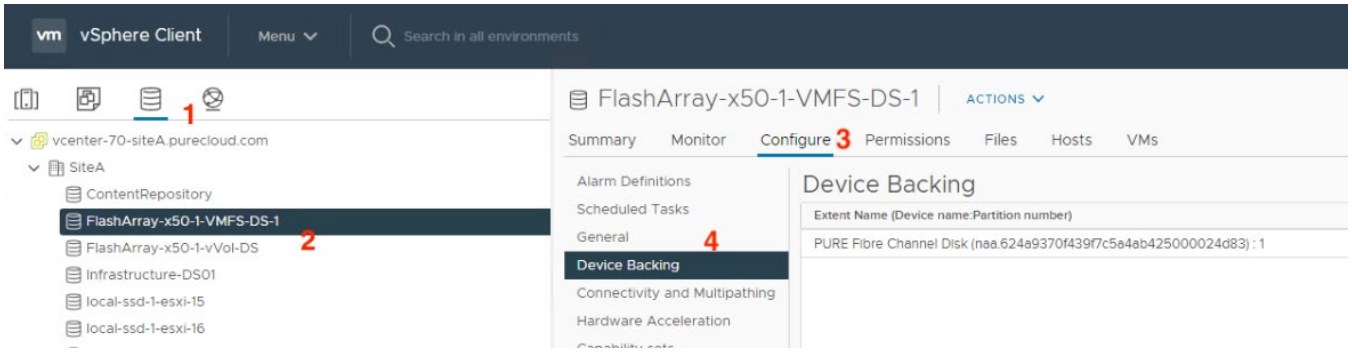


Figure 1. How to determine the device ID for your VMFS datastore.

We can now see our device ID for our VMFS datastore, which in this example is "naa.624a9370f439f7c5a4ab425000024d83."

For our vVol container:

1. Click on the **Storage** view
2. Select the vVol container
3. Click **Configure**
4. Click **Protocol Endpoints**

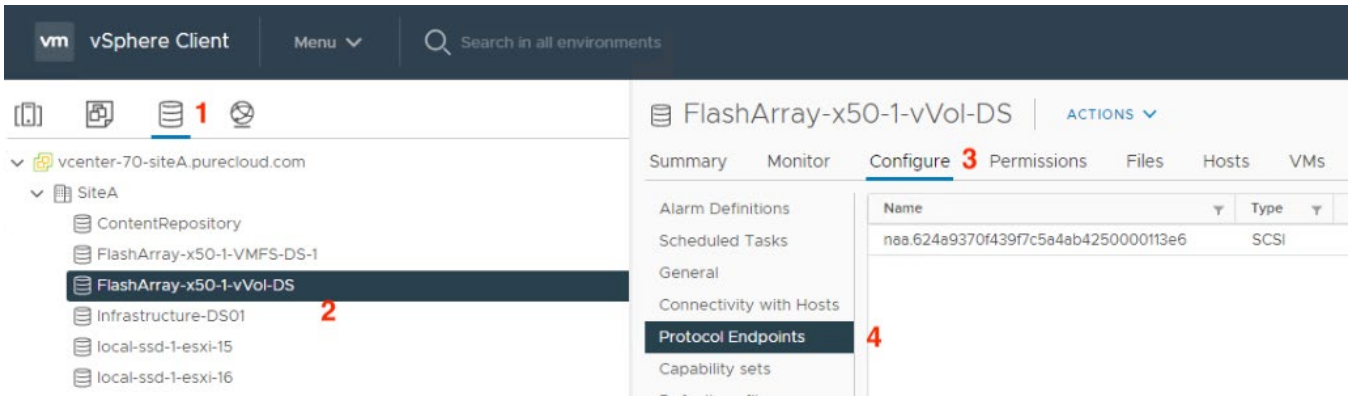


Figure 2. How to determine the device ID for your vVol container.

We can now see our device ID for our vVol container, which for this example is "naa.624a9370f439f7c5a4ab4250000113e6."

Now, we'll secure shell (SSH) to the ESXi host where our workload VM is running and look for the currently configured queue depth limit parameters.

For the VMFS datastore, run this command to check what the current values are:

```
[root@esxi-15:~] esxcli storage core device list -d naa.624a9370f439f7c5a4ab425000024d83 | grep
Depth
   Device Max Queue Depth: 32
```

We can see that our VMFS datastore has a device queue depth limit of 32 on this host by default.

```
[root@esxi-15:~] esxcli storage core device list -d naa.624a9370f439f7c5a4ab4250000113e6 | grep
Depth
   Device Max Queue Depth: 254
```

Here, we can see that our PE has a device queue depth limit of 254 on this host by default.

## Testing and Results

### Testing

In our testing, we:

1. Ensured that the workload VM was running on a VMFS datastore
2. Observed the performance from VDbench
3. Observed the status of devices in esxtop
4. Used Storage vMotion for the workload VM to our vVol container

5. Observed the performance from VDbench
6. Observed the status of devices in esxtop

VDbench is configured with these parameters for this test:

- 4k IO size
- 100% writes
- 200k IOPS
- 16 threads
- 8 JVMs

We can see that this VM is currently running on our VMFS datastore (this has a device queue depth limit of 32):

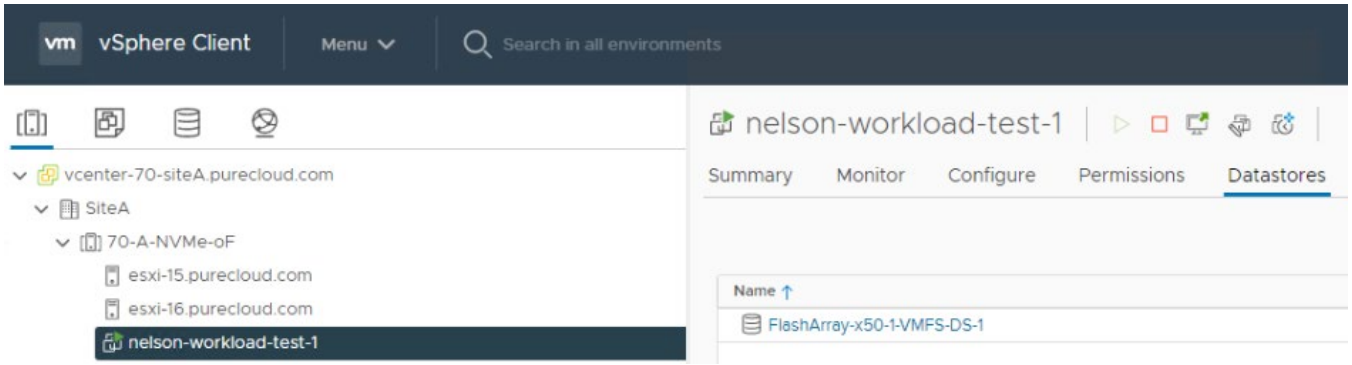


Figure 3. VM running on the VMFS datastore.

After starting our VDbench run, we can see that we're well below 200k IOPS as configured and are instead limited to about 95k IOPS with around 1.35ms of write response latency:

```
13:42:37.002 Starting RD=plumb; I/O rate: 200000; elapsed=360; For loops: xfersize=4k rdpct=0 threads=16

Sep 02, 2021  interval        i/o   MB/sec   bytes   read     resp     read    write     resp     resp queue  cpu%  cpu%
                            rate  1024**2     i/o    pct     time     resp     resp      max   stddev depth sys+u   sys
13:42:38.053         1   64682.00   252.66    4096   0.00    1.417    0.000    1.417   21.463    0.647  91.9  61.6  20.2
13:42:39.000         2   87984.00   343.69    4096   0.00    1.409    0.000    1.409   11.145    0.328 123.9  24.2  12.5
13:42:40.000         3   93412.00   364.89    4096   0.00    1.358    0.000    1.358    7.382    0.213 126.9  27.9  17.1
13:42:41.016         4   93693.00   365.99    4096   0.00    1.352    0.000    1.352    7.326    0.219 126.7  21.9  13.3
13:42:42.023         5   96373.00   376.46    4096   0.00    1.336    0.000    1.336   11.607    0.365 128.8  24.2  15.3
13:42:43.010         6   94417.00   368.82    4096   0.00    1.323    0.000    1.323    6.216    0.191 124.9  30.4  18.6
13:42:44.009         7   94339.00   368.51    4096   0.00    1.345    0.000    1.345    4.078    0.190 126.9  27.8  16.0
13:42:45.007         8   95395.00   372.64    4096   0.00    1.329    0.000    1.329    4.051    0.190 126.8  23.2  12.5
13:42:46.011         9   95295.00   372.25    4096   0.00    1.334    0.000    1.334    4.644    0.181 127.1  20.7  13.0
13:42:47.008        10   95781.00   374.14    4096   0.00    1.323    0.000    1.323    4.600    0.183 126.7  26.4  15.8
13:42:48.008        11   95357.00   372.49    4096   0.00    1.332    0.000    1.332    3.326    0.175 127.0  22.5  14.0
13:42:49.008        12   96535.00   377.09    4096   0.00    1.315    0.000    1.315    2.269    0.166 127.0  23.2  14.5
13:42:50.007        13   94742.00   370.09    4096   0.00    1.339    0.000    1.339    3.922    0.179 126.9  24.2  14.2
13:42:51.009        14   95743.00   374.00    4096   0.00    1.327    0.000    1.327    4.343    0.192 127.1  24.0  14.2
13:42:52.009        15   95227.00   371.98    4096   0.00    1.334    0.000    1.334    3.989    0.177 127.0  25.0  15.1
13:42:53.018        16   95731.00   373.95    4096   0.00    1.325    0.000    1.325    4.478    0.181 126.8  29.8  18.2
13:42:54.007        17   94784.00   370.25    4096   0.00    1.339    0.000    1.339    4.441    0.191 127.0  26.6  16.3
13:42:55.007        18   95662.00   373.68    4096   0.00    1.328    0.000    1.328    4.662    0.179 127.0  22.5  14.1
13:42:56.007        19   95709.00   373.86    4096   0.00    1.327    0.000    1.327    4.557    0.186 127.0  23.7  14.9
13:42:57.006        20   94111.00   367.62    4096   0.00    1.348    0.000    1.348    6.586    0.182 126.9  26.5  16.8
13:42:58.006        21   93880.00   366.72    4096   0.00    1.353    0.000    1.353    3.084    0.178 127.0  24.0  15.1
13:42:59.006        22   94247.00   368.15    4096   0.00    1.347    0.000    1.347    4.707    0.185 127.0  29.3  18.1
13:43:00.005        23   93594.00   365.60    4096   0.00    1.357    0.000    1.357    5.917    0.193 127.0  26.1  16.6
13:43:01.006        24   94364.00   368.61    4096   0.00    1.346    0.000    1.346    5.763    0.198 127.0  27.4  16.8
13:43:02.008        25   95071.00   371.37    4096   0.00    1.338    0.000    1.338    4.000    0.180 127.2  23.4  14.6
13:43:03.005        26   94719.00   370.00    4096   0.00    1.338    0.000    1.338    4.080    0.187 126.7  27.7  16.7
```

Figure 4. IOPS write response latency.

Why is that? Let's look at esxtop on this host for some explanation. We can see that we are queueing at the device level, and it's quite significant with 82 queued IO and 32 active IO. We can also see that KAVG (kernel average, local to the ESXi host) is .81ms. We are hitting limitations of the HBA's queue depth limit at this point in time.

```
 7:46:13pm up 5 days 21:45, 1140 worlds, 2 VMs, 17 vCPUs; CPU load average: 0.37, 0.26, 0.10

DEVICE                                 PATH/WORLD/PARTITION DQLEN WQLEN ACTV QUED %USD  LOAD   CMDS/s  READS/s WRITES/s MBREAD/s MBWRTN/s DAVG/cmd KAVG/cmd GAVG/cmd QAVG/cmd
eui.00f439f7c5a4ab4224a937500001711e          -           496     -    0    0    0 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
naa.624a9370f439f7c5a4ab4250000113e6          -           254     -    0    0    0 0.00    20.64     0.00    19.75     0.00     0.09     0.29     0.01     0.30     0.00
naa.624a9370f439f7c5a4ab42500002 4d83         -            32     -   32   82  100 3.56 100078.36     0.00 100077.91     0.00   390.93     0.30     0.81     1.11     0.80
t10.ATA_____INTEL_SSDSC2BB300G4T____          -            31     -    0    0    0 0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
```

Figure 5. Esxtop on the host results.

With our baseline VMFS testing completed, we can now move on to the next step: migrating a VM from VMFS to vVols. We will Storage vMotion the VM to our vVol datastore.

To Storage vMotion our VM from the "Hosts and Clusters" view we need to:

1. Right-click on the VM.
2. Select **Migrate.**
3. Click **Change storage only**.
4. Click **NEXT.**
5. Click to select our vVol container.
6. Click **NEXT**.
7. After reviewing for correctness, click **FINISH** to complete the operation

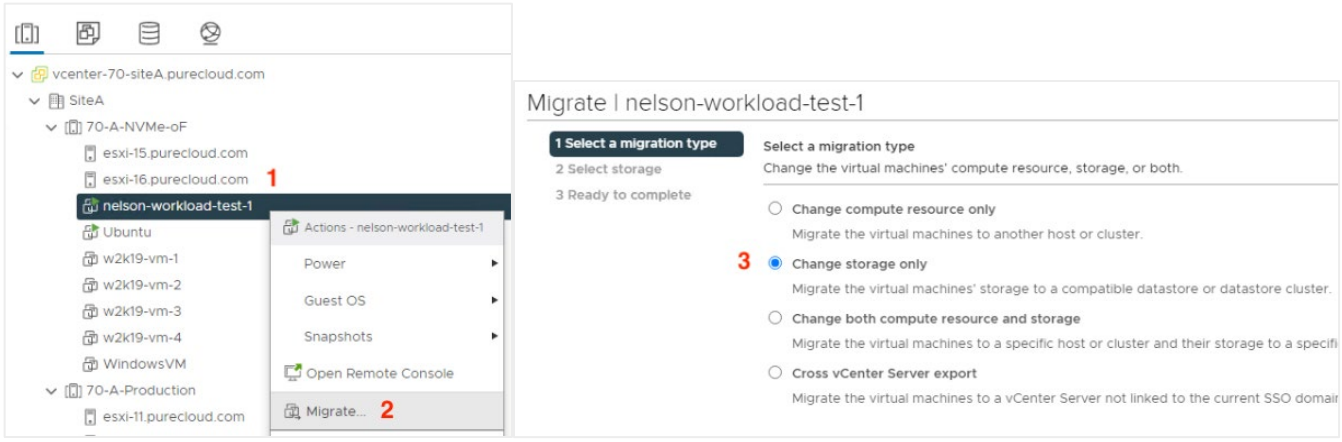


Figure 6. To Storage vMotion the VM from the "Hosts and Clusters" view.

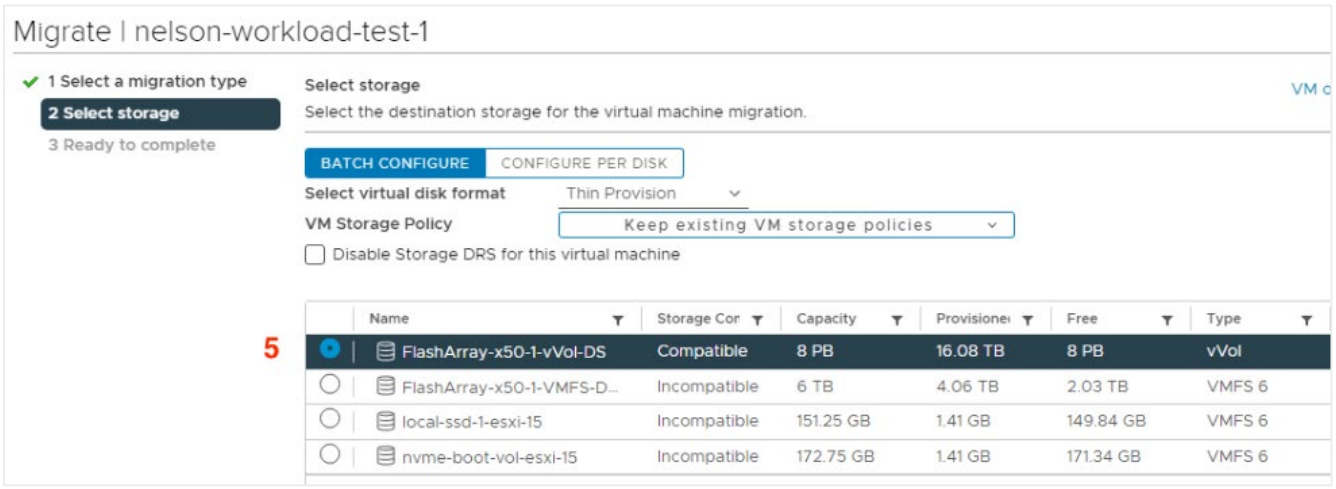


Figure 7. To Storage vMotion the VM from the "Hosts and Clusters" view.

Now that our VM is running on vVols, we'll run the same test again and observe the differences. Outside of the Storage vMotion to the vVol container, no other configuration changes were made.

Immediately in VDbench, we can see that actual IOPS more than doubled, at right around 195k, and write response is down to about .66ms, cutting that latency by about half vs VMFS in Figure 8:

```
14:06:06.320 All slaves are now connected
14:06:09.002 Starting RD=plumb; I/O rate: 200000; elapsed=360; For loops: xfersize=4k rdpct=0 threads=16

Sep 02, 2021  interval        i/o   MB/sec   bytes   read     resp     read    write     resp     resp queue  cpu%  cpu%
                            rate  1024**2     i/o    pct     time     resp     resp      max   stddev depth sys+u   sys
14:06:10.078        1   41374.00   161.62    4096   0.00    1.655    0.000    1.655  296.414    6.086  69.2  60.2  37.1
14:06:11.014        2  105507.00   412.14    4096   0.00    1.176    0.000    1.176   36.904    1.445 123.5  86.6  59.4
14:06:12.019        3  179536.00   701.31    4096   0.00    0.702    0.000    0.702    7.689    0.223 126.0  54.4  33.3
14:06:13.024        4  176062.00   687.74    4096   0.00    0.710    0.000    0.710    9.411    0.252 125.1  40.6  23.0
14:06:14.013        5  183409.00   716.44    4096   0.00    0.687    0.000    0.687   17.239    0.314 125.9  52.1  30.5
14:06:15.008        6  192250.00   750.98    4096   0.00    0.651    0.000    0.651    6.061    0.196 125.2  58.5  33.2
14:06:16.008        7  192978.00   753.82    4096   0.00    0.651    0.000    0.651    3.719    0.183 125.7  53.0  34.5
14:06:17.010        8  190945.00   745.88    4096   0.00    0.660    0.000    0.660    4.908    0.195 125.9  48.8  30.8
14:06:18.009        9  190679.00   744.84    4096   0.00    0.659    0.000    0.659    4.720    0.195 125.7  43.4  27.5
14:06:19.007       10  192219.00   750.86    4096   0.00    0.654    0.000    0.654    6.040    0.198 125.7  49.2  32.3
14:06:20.009       11  186847.00   729.87    4096   0.00    0.674    0.000    0.674    4.364    0.202 125.9  48.3  30.6
14:06:21.008       12  191426.00   747.76    4096   0.00    0.657    0.000    0.657    3.113    0.186 125.9  53.6  33.2
14:06:22.007       13  186571.00   728.79    4096   0.00    0.674    0.000    0.674    4.692    0.203 125.8  54.2  34.5
14:06:23.008       14  191915.00   749.67    4096   0.00    0.656    0.000    0.656    4.029    0.194 125.9  50.6  34.1
14:06:24.007       15  186040.00   726.72    4096   0.00    0.677    0.000    0.677    4.851    0.204 125.9  45.8  30.1
14:06:25.019       16  179763.00   702.20    4096   0.00    0.700    0.000    0.700   11.967    0.271 125.8  44.1  27.3
14:06:26.008       17  182665.00   713.54    4096   0.00    0.690    0.000    0.690    5.482    0.228 126.0  48.8  30.0
14:06:27.007       18  187387.00   731.98    4096   0.00    0.671    0.000    0.671    7.170    0.215 125.8  57.1  38.3
14:06:28.006       19  190571.00   744.42    4096   0.00    0.661    0.000    0.661    5.379    0.193 125.9  50.4  33.5
14:06:29.007       20  191351.00   747.46    4096   0.00    0.658    0.000    0.658    3.530    0.190 125.9  53.4  34.9
14:06:30.007       21  191606.00   748.46    4096   0.00    0.657    0.000    0.657    3.045    0.183 125.9  49.7  30.8
14:06:31.005       22  194076.00   758.11    4096   0.00    0.648    0.000    0.648    4.380    0.183 125.7  58.1  36.7
14:06:32.005       23  194495.00   759.75    4096   0.00    0.647    0.000    0.647    3.854    0.183 125.8  51.9  33.9
14:06:33.006       24  193397.00   755.46    4096   0.00    0.651    0.000    0.651    5.388    0.198 125.9  46.2  29.0
14:06:34.006       25  196510.00   767.62    4096   0.00    0.640    0.000    0.640    3.736    0.187 125.7  51.4  32.5
14:06:35.005       26  195725.00   764.55    4096   0.00    0.643    0.000    0.643    3.509    0.180 125.9  47.0  32.6
```

Figure 8. Storage vMotion results.

Looking in esxtop on our host, we can see the reason why: there is no longer queueing at the device level and our KAVG latency has significantly dropped to .08ms, down from .81ms on VMFS in Figure 9:

```
 8:07:27pm up 5 days 22:07, 1140 worlds, 2 VMs, 17 vCPUs; CPU load average: 0.64, 0.17, 0.07

DEVICE                                 PATH/WORLD/PARTITION DQLEN WQLEN ACTV QUED %USD  LOAD   CMDS/s  READS/s WRITES/s MBREAD/s MBWRTN/s DAVG/cmd KAVG/cmd GAVG/cmd QAVG/cmd
eui.00f439f7c5a4ab4224a937500001711e                    -   496     -    0    0    0  0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
naa.624a9370f439f7c5a4ab4250000113e6                    -   254     -   98    0   38  0.39 194739.04     0.00 194737.67     0.00   760.69     0.43     0.08     0.51     0.04
naa.624a9370f439f7c5a4ab425000024d83                    -    32     -    0    0    0  0.00     0.46     0.00     0.00     0.00     0.00     0.59     0.03     0.61     0.00
t10.ATA_____INTEL_SSDSC2BB300G4T____                    -    31     -    0    0    0  0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00     0.00
```

Figure 9. Storage vMotion results.

## Results

While these results are dramatic, keep in mind that this is a single VM on a single host. Despite that, we still saw a significant performance improvement. Although not every environment is the same, it is possible to see even greater improvements when running concurrent workloads with more hosts and VMs.

There are two key pieces of information to break down here: the difference in latency and IOPS observed while on VMFS vs vVols in VDbench.

Looking at write latency first, we dropped from an average of just under 1.4ms on VMFS in Figure 4 to just above .7ms on vVols in Figure 8, which is about a 50% drop in observed latency at the Guest OS level:

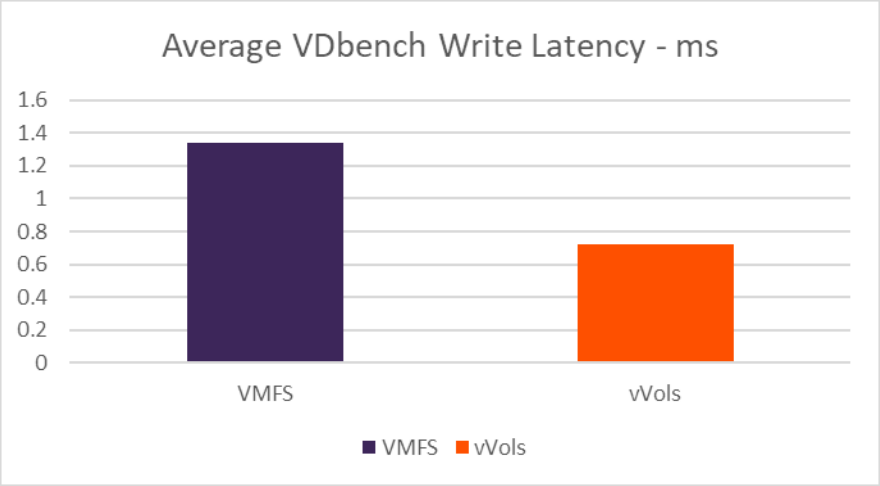


Figure 10. Average VDbench write latency.

Lastly, we can see that IOPS almost doubled, going from just under 90k on VMFS in Figure 4, to just above 180k on vVols in Figure 11:

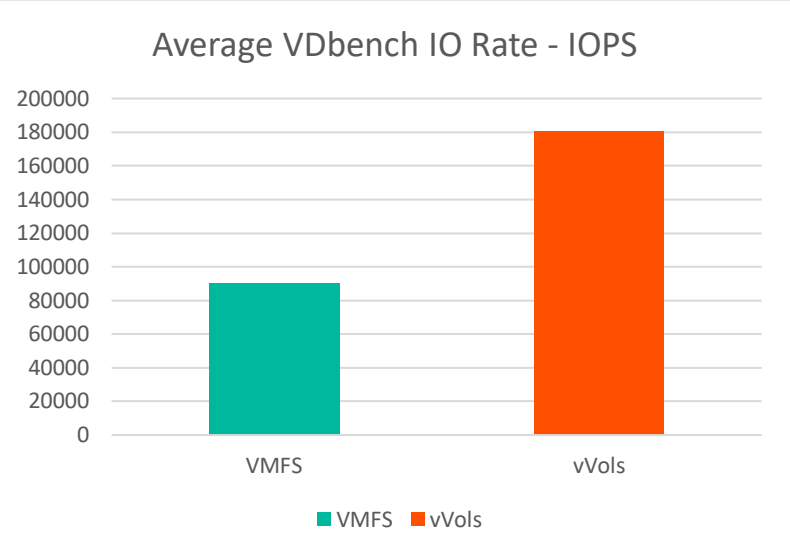


Figure 11. Average VDbench IO rate.

## Conclusion

With Emulex's new FC HBA drivers, VMware administrators no longer need to concern themselves with manual PE queue depth limit configuration. Additionally, the performance you have come to expect from your vVol environment, even for demanding workloads, comes with something else you expect from vVols: simplicity of administration. When paired with Pure Storage's high-performance FlashArray, Emulex's FC HBAs enable your entire vVol stack to have higher performance, lower latency, and a simpler configuration.

## Additional Resources

- Broadcom 12.8.534.0 Driver Documentation
- Pure Storage vVol Platform Guide
- Cody Hosterman's Queue Depth blog post
- Emulex VMware Compatibility Guide listing



Pure Storage, Inc.
650 Castro Street, #400
Mountain View, CA 94041