vSAN Proof of Concept: vSAN Performance Testing

VMware Storage

Table of contents

vSAN Proof of Concept: vSAN Performance Testing

Performance Testing

As with all enterprise-class storage solutions, there are many variables that may impact performance: type of hardware, network infrastructure, cluster design, and workload performance characteristics, all contribute to performance testing results. Often, performance is limited by bottlenecks in the system. Removing one potential bottleneck (such as networking) will mean the performance may be constrained by another (such as CPU). Therefore it is imperative that the system is designed and sized correctly.

For more information, a good overview on troubleshooting vSAN performance is provided here:

https://core.vmware.com/resource/troubleshooting-vsan-performance
https://core.vmware.com/resource/performance-recommendations-vsan-esa

Hardware Design & Sizing for High Performance

For the highest levels of performance, undivided and unshared resources, including CPU, memory, and network are required. Of course, in the real world, there will be compromises made (for better efficiencies, or simply cost). For the best possible performance, consider the following design parameters:

- Low or no CPU and memory overcommitment
- Appropriate host power management settings
- Use only NVMe or better devices (with multiple disk groups per host for vSAN OSA): an overcommitted PCI-Bus should be avoided
- Unconstrained network bandwidth, wherever possible
- Utilize switches with deep buffers
- Ensure that the network is designed to avoid overcommitment

Physical Storage Device Choice

Consider choosing devices with highest possible performance class, as this will generally be the first bottleneck in the system. For vSAN OSA, an all-flash vSAN design is necessary to achieve the lowest possible latencies.

SSD or NVMe drives have different levels of performance, depending on how they are made. To illustrate, an NVMe drive performance class hierarchy is shown below. Note this is highly dependent on the NVMe vendor:

NVMe 3D Xpoint > NVMe MLC high spec > NVMe ‘low spec’ endurance

The highest performance configuration is with fast (certified) NVMe devices and vSAN ESA.

In high-performance vSAN OSA configurations, capacity disks are usually chosen from (no more than) one category lower to the caching tier to achieve an ideal balance of latency and throughput during de-staging phase.

Network Fabric and Hardware Design Choice

The next biggest bottleneck is usually the network. For distributed systems, network design a critical factor, directly affecting system stability and IO performance. The network design should be careful to factor in the fact that storage data (compared to other data) is highly latency sensitive. For instance, designs should include switches with deep buffers and the switch configurations should steer clear of anything that could adversely affect storage traffic (such as Quality of Service). Moreover, switches with backplanes that cannot support full link utilization among all ports are not recommended. Finally, devices that introduce bottlenecks to an upstream switching device for cross-port communications (such as fabric extenders), should not be considered.

See Appendix A for guidance on calculating switch buffers

Synthetic vs. Real-Workload Testing Overview

In an ideal world, running the actual workload (to be productionized) on a testbed would give us the most useful results. However, this is often not possible. Here, synthetic benchmark tests can be a useful tool to analyze storage performance, if used correctly. Such tests require that you understand the workload profile to be tested, and for accuracy, these should be derived from real world application analysis.

Data can be gathered by using several methods. Estimates could be made from performance views in vCenter (see the vSAN Management, Monitoring and Hardware Testing guide) or Aria Operations (formerly vRealize Operations). Other tools include Live Optics (free to use, see example below) or OneIQ.

Once the data from existing workloads has been gathered, we can then proceed with creating synthetic tests to simulate workloads. Note that care needs to be taken when simulating workloads to ensure that we are testing the system (as a whole) and not just individual components. Following this logic, it would be naive to take the storage data obtained and apply it to benchmarks on single hosts. Indeed, this is the case for all distributed systems, namely that a holistic approach needs to be employed and performance tests need to be both scalable and homogenous. VMware has a free tool: ‘HCIbench’ which is designed to test distributed storage systems (essentially an automation wrapper for the open-source FIO and Vdbench testing utilities). See the next section for detailed information on using HCIbench.

At a high level, the steps involved in performance testing are then:

- Obtain metrics from the existing workloads
- Simulate workloads using synthetic tests
- Change Storage Policies and/or vSAN services as needed
- Compare results

Note that if no existing metrics are available, we will have to create some tests based on assumptions. Some of these could include an “industry standard” 4k/70% read/random IO test – or test the limits of the system with a large block size and 100% read or writes. For the second option, there are better methods to gauge the *maximum* performance of the system: see the ‘Maximum Performance Test’ section below.

Moreover, there may be some guidance provided by whitepapers (by VMware or the software/database vendor) on storage configuration for optimal performance and reliability. These recommendations should be considered both when configuring the size of the workload (e.g. the quantity of disks attached to a VM and the use case for those disks by the application), as well as the vSAN Storage Policy that is applied to the VM/virtual disk objects.

For example, if testing the performance of a SQL server solution, follow the VMware whitepaper on SQL best practices for vSAN (such as using a RAID-1 storage policy and setting the Object Space Reservation of 100% for log disks).

Terminology

Typical to storage performance analysis, the terms below are frequently used:

Block size or IO size (usually in KB) is the read or write request size sent from an application to the storage layer

Bandwidth Usually expressed in MB/s (Megabytes per second), is a measure of data transfer capacity

IOPS (IOs per second) is a measure of throughput, defined as the bandwidth (KB/s) divided by the IO size in KB. Thus: IOPS = 1024 x Bandwidth (MB/s) / IO size (KB)

IO latency (usually in ms) reflects the time delay for each IO to be completed

Outstanding IO or IO queues is the number of IOs waiting to be committed from the operating system down to the storage layer

Performance Testing Using HCIBench

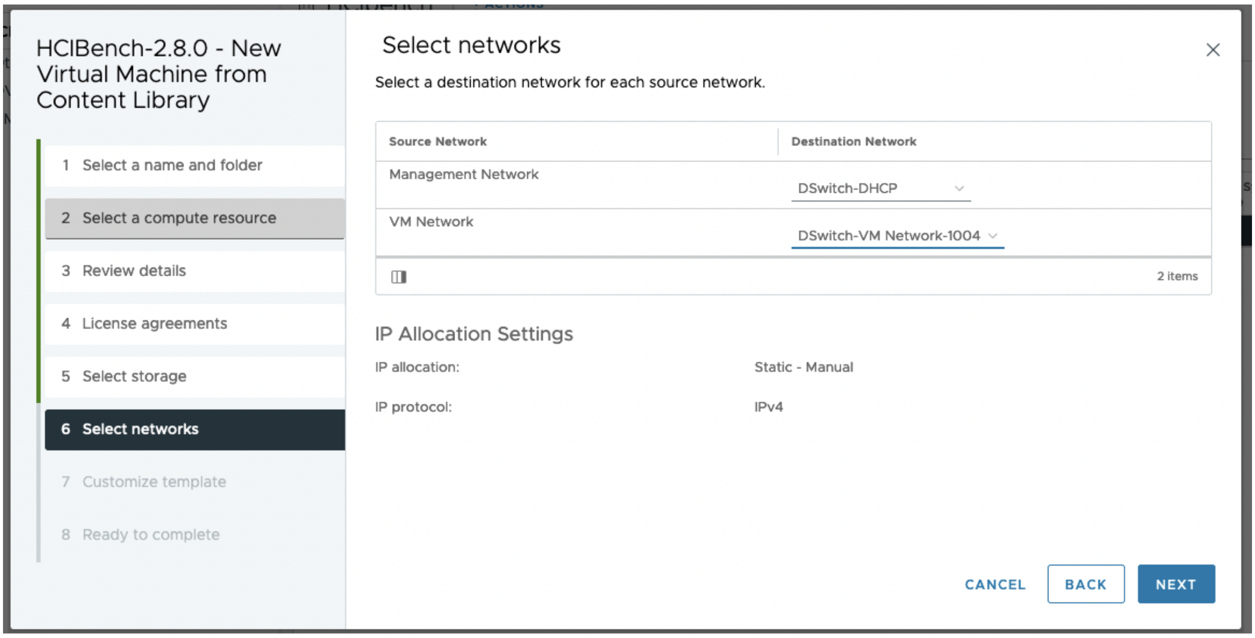
HCIBench is a free tool that automates the configuration and distribution of workload generator VMs. Two synthetic engines are included for the worker VMs, the open-source FIO and Vdbench. Any storage system (not just vSAN) can be tested and compared.

HCIBench (and documentation) is available here: https://labs.vmware.com/flings/hcibench

Configuring HCIBench

Firstly, deploy the HCIBench OVA. The options here are self-explanatory (consult the documentation found on the flings page if in any doubt):

In particular, pay attention to the network selection. There are two networks needed, the management network (to manage HCIBench itself) and the network used to deploy the test VMs (‘VM Network’).



Select networks

Select a destination network for each source network.

Source Network	Destination Network
Management Network	DSwitch-DHCP
VM Network	DSwitch-VM Network-1004

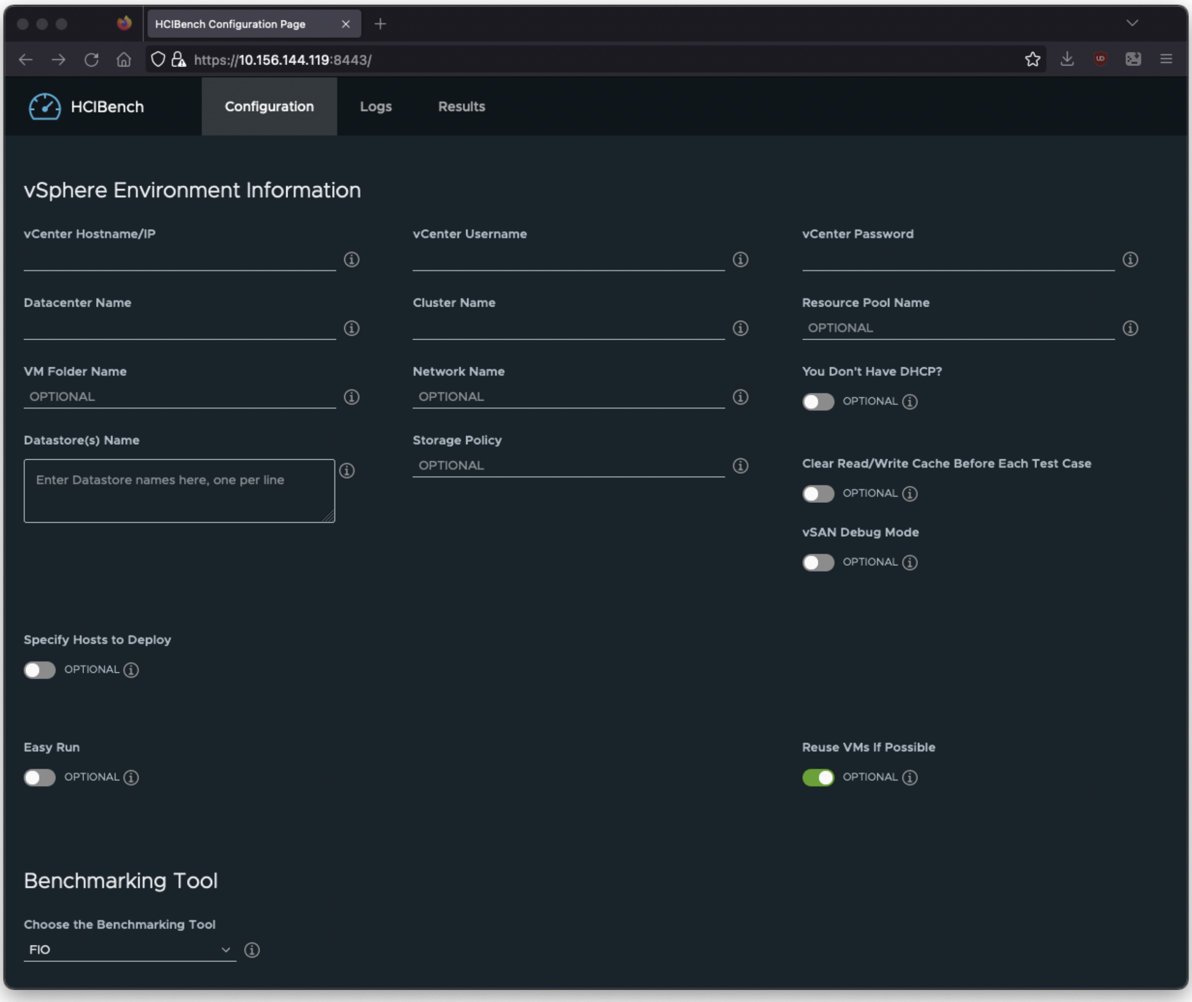
2 items

IP Allocation Settings

IP allocation: Static - Manual

IP protocol: IPv4

After deployment, navigate to http://<Controller VM IP>:8080/ to configure HCIbench



vSphere Environment Information

vCenter Hostname/IP

vCenter Username

vCenter Password

Datacenter Name

Cluster Name

Resource Pool Name: OPTIONAL

VM Folder Name: OPTIONAL

Network Name: OPTIONAL

You Don't Have DHCP?: OPTIONAL

Datastore(s) Name: Enter Datastore names here, one per line

Storage Policy: OPTIONAL

Clear Read/Write Cache Before Each Test Case: OPTIONAL

vSAN Debug Mode: OPTIONAL

Specify Hosts to Deploy: OPTIONAL

Easy Run: OPTIONAL

Reuse VMs If Possible: OPTIONAL

Benchmarking Tool

Choose the Benchmarking Tool: FIO

There are three sections to consider:

- vSphere information
- Test VM setup
- Workload Config

Either FIO (the default) or Vdbench can be chosen as the Benchmarking Tool for the worker VMs. Here, we recommend using FIO, due to the exhaustive list of parameters that can be set. Pre-defined parameter files can be uploaded to HCIbench (a wide variety of options are available, such as different read/write block sizes). For a full list of FIO options, consult the FIO documentation.

Below we walk through the considerations on defining the workload configuration:

Working Set

Our first consideration is how big to make our workload. This will be determined by the number of worker VMs per host, and then the number and size of the disks on each worker VM. This is our 'working set', and for best performance, the working set should be mostly in the cache.

With this in mind, let's illustrate this with an example:

Consider a four-node vSAN OSA cluster with one 400GB SSD per node. This gives the cluster a total cache size of 1.6TB (4x400GB). For a hybrid cluster, the total cache available in vSAN is split 70/30 for read cache and write cache, thus we have 1120GB available. Therefore, to fit the workload within the cache, the total capacity of all VMDKs used for I/O testing should not exceed 1.6TB (or 1,120GB for hybrid).

Then, if we define our workload with four VMs per host, each VM having 5 x 10GB VMDKs, we have a total working set size of 5 x

10GB x 4VMs x 4Hosts = 800GB, which fits within our cache of 1.6TB. **Note**: the maximum cache allocatable per cache disk in vSAN OSA is 1.2TB. If your cache disk size is greater, use this value for the size of each cache disk.

Be mindful that increasing the number of VMs, whilst spreading the load across the cluster, will consume both CPU and memory of the host. Similarly, increasing the number of virtual disks per VM will increase the number of queues available to use (but too many may overwhelm the resources available).

Of course, a valid test scenario is to look at the effects of overload the cache (as a worst-case scenario). In this case, we take the values above as the minimum (VMs and number/size of virtual disks).

Prepare Virtual Disk Before Testing

To achieve a 'clean' performance run, the disks should be wiped before use. To achieve this, select a value for the 'Prepare Virtual Disk Before Testing'. This option will either zero or randomize the data (depending on the selection) on the disks for each VM being used in the test, helping to alleviate a first write penalty during the performance testing phase. We recommend that the disks are randomized if using the Deduplication & Compression feature.

Storage Policy

Here, we assign a storage policy to our test run data. **It is recommended to create and use a RAID-5 storage policy for vSAN ESA**. This is due to the way data is managed in vSAN ESA; each VM disk will have a performance and capacity leg. The performance leg is always a RAID-1 mirror (with the capacity leg following the placement defined in the storage policy).

If no policy is defined, the default is to use the ‘vSAN default storage policy’, which is a RAID-1 mirror.

Warm up period

As a best practice, performance tests should include at least a 15-minute warm-up period. Also, keep in mind that the longer testing runs the more accurate the results will be. Warm-up period is used mainly in hybrid solutions.

Testing Runtime

HCIbench tests should be configured for *at least* one hour to observe the effects of de-staging from the cache to the capacity tier. Runtime is defined by the amount of block changes in the caching tier and chosen workload load profile. Small blocksizes required more time to achieve a total block change on the cache versus large blocksizes.

The test run duration can make a big difference to the results. Therefore, consider long test runs to warm up and make full use of the cache, see multiple de-staging phases and to identify max/95%/avg and sustained workload.

Results

After testing is completed, you can view the results at **http://<Controller VM IP>:8080/results** in a web browser. A summary file of the tests will be present inside the subdirectory corresponding to the test run. To export the results to a ZIP file, click on the 'save result' option on the HCIbench configuration page (and wait for the ZIP file to be fully populated).

As HCIbench is integrated with the vSAN performance service, the performance data can also be reviewed within the vCenter HTML5 UI, under **[vSAN cluster] > Monitor > vSAN > Performance**.

‘Easy Run’

Although 'Easy Run' can be selected, we recommend explicitly defining a workload pattern to ensure that tests are tailored to performance requirements.

Blocksize

It is important to match the block size of the test to that of the workload being simulated, as this will directly affect the throughput and latency of the cluster. Therefore, it is paramount that this information be gathered before the start of the tests (for instance, from a Live Optics assessment).

If using FIO, block sizes can be set separately for reads and writes (using the blocksize parameter). Moreover, a range of sizes can be specified by using the bssplit parameter. For more information, consult the FIO documentation.

Sequential and Random workloads

Sustained sequential write workloads (such as VM cloning operations) will simply fill the cache in vSAN OSA configurations, potentially creating latency for further writes, until they are de-staged to the capacity tier. This is especially relevant in a hybrid environment, where the performance will be largely dependent on the speed of spinning disks. The same is true for sustained sequential reads. If the block is not in the cache, it will have to be fetched from the spinning disk. Mixed workloads will benefit most from vSAN's OSA caching design.

Other Considerations

vSAN OSA hybrid clusters are defined by the use of flash drives only for the cache tier. Magnetic drives are used for the capacity tier. Unsurprisingly, the performance between flash (the cache tier) and magnetic drives (capacity tier) is significant: flash drives are usually an order of magnitude faster Therefore, if reads can be served by the caching tier, then performance will be the greatest. The Ideal approach then is to have upwards of 90% of all IO served from the caching tier.

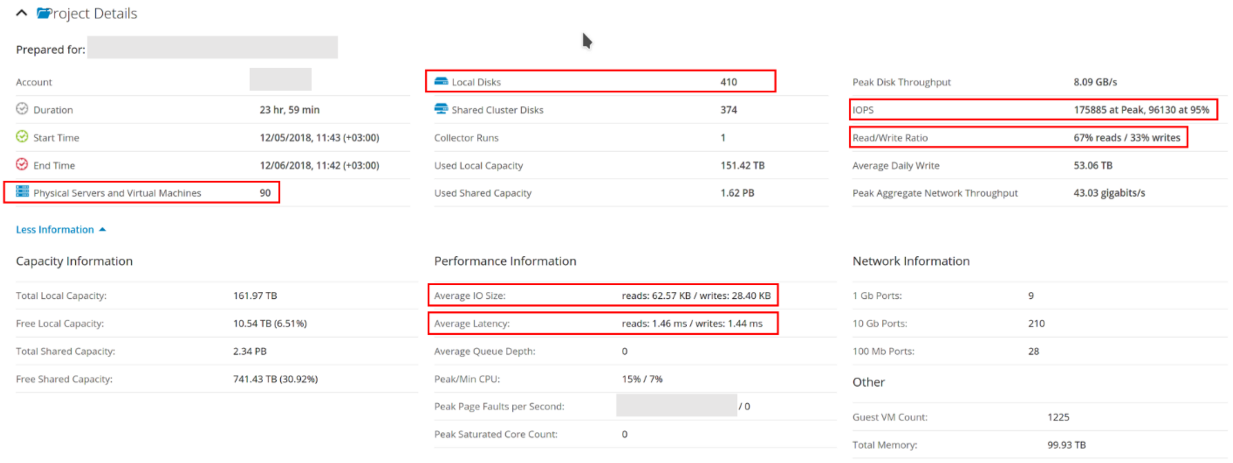
Also consider some tests that run over the size of the cache, to cover burst and overloaded scenarios.

HCIBench Tests using Liveoptics Data

Existing Workload Analysis using Live Optics

The relevant metrics are easily identifiable in the output from a performance auditing tool such as Live Optics:

Example LiveOptics Report:



Project Details

Prepared for: [redacted]

Account	[redacted]	Local Disks	410	Peak Disk Throughput	8.09 GB/s
Duration	23 hr, 59 min	Shared Cluster Disks	374	IOPS	175885 at Peak, 96130 at 95%
Start Time	12/05/2018, 11:43 (+03:00)	Collector Runs	1	Read/Write Ratio	67% reads / 33% writes
End Time	12/06/2018, 11:42 (+03:00)	Used Local Capacity	151.42 TB	Average Daily Write	53.06 TB
Physical Servers and Virtual Machines	90	Used Shared Capacity	1.62 PB	Peak Aggregate Network Throughput	43.03 gigabits/s

Less Information ▲

Capacity Information

Total Local Capacity:	161.97 TB
Free Local Capacity:	10.54 TB (6.51%)
Total Shared Capacity:	2.34 PB
Free Shared Capacity:	741.43 TB (30.92%)

Performance Information

Average IO Size:	reads: 62.57 KB / writes: 28.40 KB
Average Latency:	reads: 1.46 ms / writes: 1.44 ms
Average Queue Depth:	0
Peak/Min CPU:	15% / 7%
Peak Page Faults per Second:	[redacted] / 0
Peak Saturated Core Count:	0

Network Information

1 Gb Ports:	9
10 Gb Ports:	210
100 Mb Ports:	28

Other

Guest VM Count:	1225
Total Memory:	99.93 TB

When collecting performance data, long-term capture periods provide more accurate statistics for peak IO load and 95%-percentile performance. Performance statistics should not be collected for less than 24 hours if possible, and 7-10 days is ideal. In any case, it is critically important to capture performance statistics that highlight peak load during working hours, together with any IO load generated by off-hours operations (such as batch data processing or system backup jobs). Note also that LiveOptics is not limited to just virtualized environments; physical hosts can be included in the data capture.

Using Live Optics data in HCIBench

From the previous section, refer to the Liveoptics example (see 'Existing Workload Analysis using Liveoptics'). From this data, we can identify the following details to define a workload for testing with HCIbench:

- Number of VMs
- Number of Disks
- IOPS peak
- Read/write ratio
- IO block size
- IO latency

We can then create a workload profile for FIO, using the data from the example and some simple calculations:

Number of VM disks:

```
#disks / #VMs = 410 / 90 which is approx. 5 disks per VM
```

IOdepth or outstanding IO:

```
Total outstanding IO = peak IOPS (sum of reads + writes) x (reads + write latency)
                     = 175885 x 0.0029sec
                     = 510

Outstanding IO per VMDK = 510 / 410 VMDK = 1.2 □ round up to IO depth of 2 per VMDK
```

This provides the parameters for the FIO custom profile definition for HCIbench:

```
[global]
runtime=3600
time_based=1
rw=randwrite
rwmixread=67
percentage_random=100
blocksize=62k,28k
ioengine=libaio
buffered=0
direct=1
fsync=1
group_reporting
log_avg_msec=1000
continue_on_error=all
size=100%
iodepth=2

[job0]
filename=/dev/sda

[job1]
filename=/dev/sdb
```

When testing any storage solution, be sure to set a sufficiently long run-time to accurately assess the performance characteristics of the system. Short performance benchmark runtimes often present inaccurate metrics that are artificially enhanced by caching performance but cannot be sustained during longer term operations.

Maximum Performance Test

Storage Solutions most often use a cache function to buffer IO sent to physical drives. Any test that probes the maximums of what a storage solution can attain should stress the limits of this caching tier.

The recommended procedure is to look at the amount of parallel outstanding IO (or *io_depth* in FIO) against IO latency. FIO (with the *latency_target* parameter) can gradually ramp up the amount of outstanding IO until a set latency figure is reached. Thus, we have a method to systematically overload the setup without overwhelming the hardware.

In the example FIO parameter file below, we set a target of 50ms IO latency. FIO will sample the latency within the specified *latency_window*, which is set to 10s (thus FIO will ramp up IO every five seconds). We set the I/O percentile (latency_percentile) to 95% to disregard any sudden spikes.

The other parameters include:

- Read/write ratio 70/30%; 100% random; 4k block reads and writes
- Unbuffered and verified
- The outstanding IOs to start at 1, with a max of 256 for each disk

The FIO parameter file can also be found on Github:
https://raw.githubusercontent.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/main/VCF/fio_profiles/max_perf.fio

```
[global]
runtime=3600
time_based=1
rw=randrw
rwmixread=70
percentage_random=100
```

```
blocksize=4k,4k
ioengine=libaio
buffered=0
direct=1
fsync=1
group_reporting
log_avg_msec=1000
continue_on_error=all
iodepth=256
iodepth_low=1
latency_target=50ms
latency_window=10s
latency_percentile=95

[job-sda]
filename=/dev/sda

[job-sdb]
filename=/dev/sdb
```

Here we use two disks on each test VM (more disks can be added to mimic production) with around 4-16VMs per host, depending on how dense the workloads need to be. Over the course of the IO test, FIO will increase the number of outstanding IO and thus the latency will increase until the target latency (and possibly slightly over) is reached. Once the target is reached, FIO will end the test.

Note: when testing against traditional storage systems, incorporate one LUN per VM (with all virtual disks on the same LUN). This will ensure a like-for-like test.

Steady State Test

Here, we fix the number of IO queues to show the effects of a steady state. A good approach is to run the maximum performance test (see above) and obtain a value of outstanding IO that corresponds to an acceptable write latency figure.

In the example below, we have two virtual disks with the number of IO queues (iodepth parameter) set to 9. Also:

- Blocksizes are split between several values (bssplit parameter)
- The read/write ratio is set to 70%
- Data is 80% random

The FIO parameter file can also be found on Github: https://raw.githubusercontent.com/vmware-tanzu-experiments/vsphere-with-tanzu-proof-of-concept-samples/main/VCF/fio_profiles/steady_state.fio

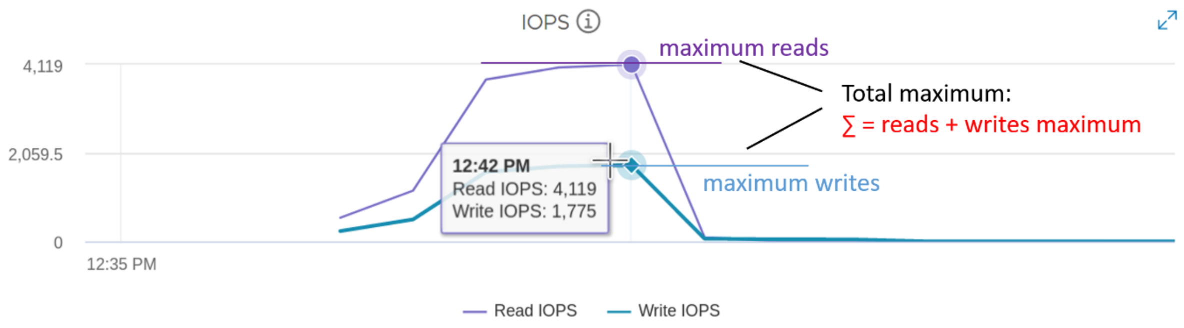
```
[global]
runtime=3600
time_based=1
rw=randrw
rwmixread=70
percentage_random=80
bssplit=4k/30:64k/30:32k/40
ioengine=libaio
buffered=0
direct=1
fsync=1
group_reporting
log_avg_msec=1000
continue_on_error=all

[job0]
filename=/dev/sda
iodepth=9

[job1]
filename=/dev/sdb
```

```
iodepth=9
```

APPENDIX A: IO LOAD DEFINITIONS

Peak Workload

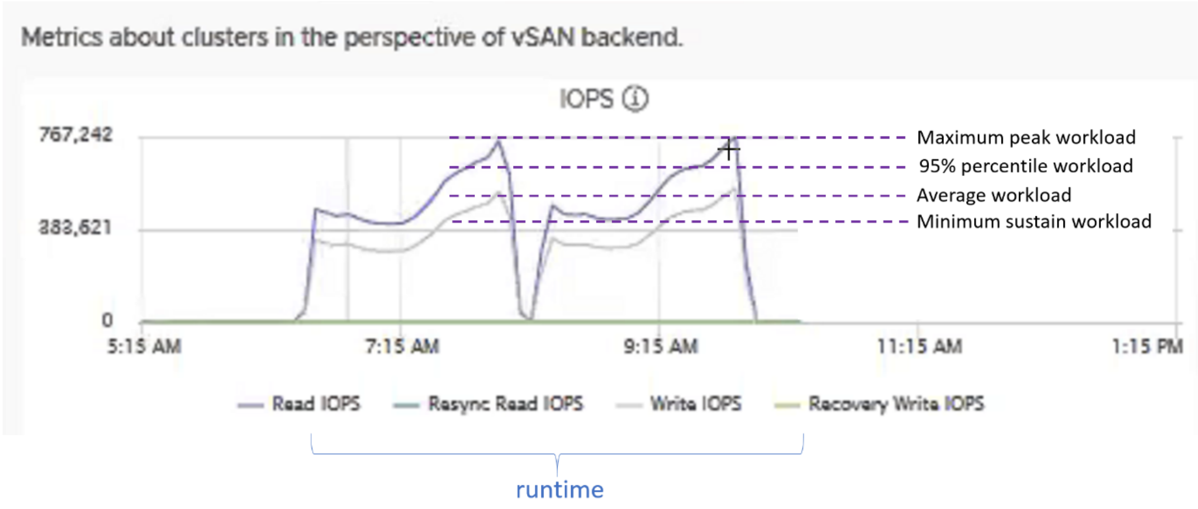
Peak workload performance reflects the observed maximum of a storage system for all hosts/workloads participating in a performance test. Spikes to peak workload performance are generally associated with IO intensive application operations.



Average Performance

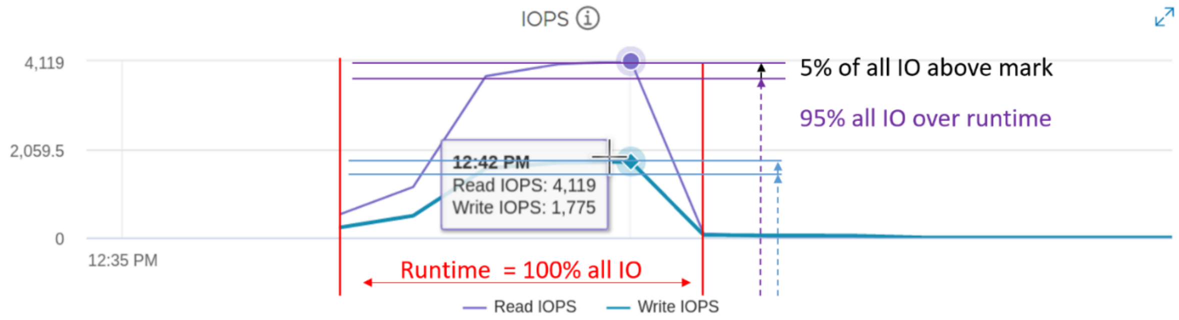
Average performance provides the total average IO performance across all VMs within a cluster or host and time frame and is less ideal for workload profile definition. IO burst or unusually high IO outlier workloads may be obscured by other more normalized workloads contributing to the average.

Peak and 95th percentile workload are the most reliable values to be used to develop an IO profile for synthetic benchmark testing, and most effectively demonstrate the cluster's sustainable workload performance when tested for an adequate period of time (> 1hr).



95th Percentile Workload

95% percentile workload performance identifies 95% of all IO processed in a specific time frame (capture time):


IOPS ⓘ
4,119
2,059.5
0
12:35 PM
12:42 PM
Read IOPS: 4,119
Write IOPS: 1,775
5% of all IO above mark
95% all IO over runtime
Runtime = 100% all IO
Read IOPS
Write IOPS

APPENDIX B: Calculating Ideal Switch Buffers

Example: 25Gbit/s link speed, RTT latency with 1ms between hosts

- 25Gbit/s for TX & RX full duplex = 2x 3125 MByte/s at line speed
- 1ms round trip time
- 1x TX 3125MByte/s x 0.001 seconds = 3.125 MByte buffer per port

Assuming we have multiple hosts attached to one switch, multiply by the total ports consumed to obtain total required buffer size.

Example: 48port switch, 12 hosts consuming the same switch (assumptions from prior example remain unchanged)

- 12 x 3.125 MByte = 37.5MByte switch buffers (TX) minimum required

Note that some models of switch may be configured in ways that impact the amount of buffer memory available to any given port. To ensure that a switch meets these estimated requirements,
consult the switch documentation, review the switch configuration, and if necessary, obtain assistance from the switch vendor.

Inter-switch link/uplink (ISL) capacity must be considered in cases where vSAN traffic will flow between hosts connected to distinct physical switches.

For example, continuing the above example with 12 hosts, we can estimate the peak level of possible traffic northbound to leaf to spine/ super spine or core switch:

- Extremely high IO may result in full link utilization for RX/TX on each host
- The bandwidth requirement for the inter-switch link may be found by multiplying:

Number of hosts x their network link speed x the number of ports per host; in this example 12 x 25Gbit x 2 = 600Gbit/s bandwidth (TX+RX) capacity required to support theoretical peak utilization.

Note that though inter-switch links may be overcommitted, IP latency resulting from congestion during peak utilization will directly increase IO latency for vSAN.

Additionally, vSAN cluster designs such as 2-node, stretched cluster, and use of manually defined fault domains can introduce additional network hops and traffic requirements that result in additional IO latency; take limitations regarding factors such as inter-site links into account when planning to test performance in these scenarios.