
CONTENTS

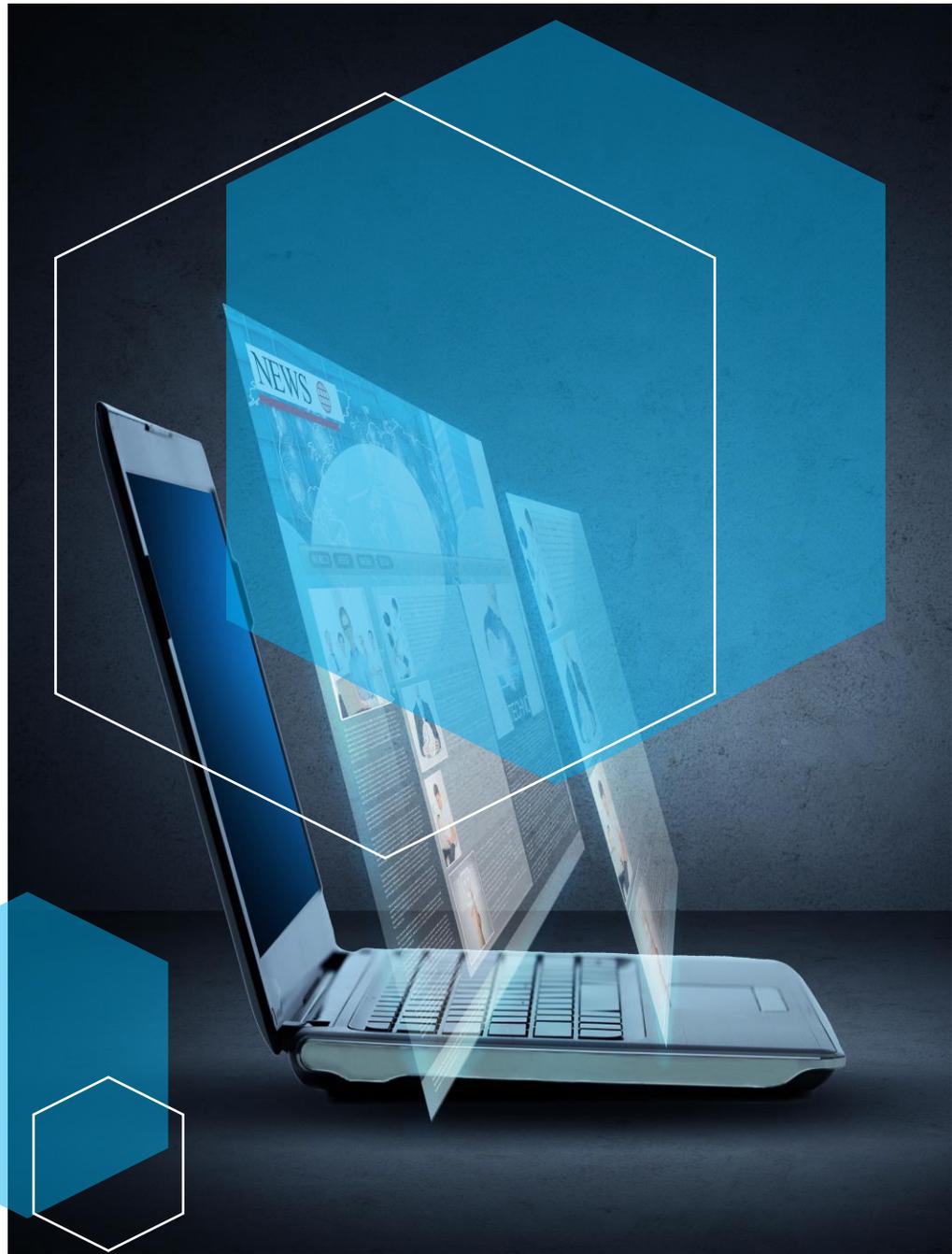
Bottlenecks for Performance

Impact of Provisioning Technologies on Performance

Iometer Advantages and Disadvantages

Recommended Configuration of Iometer

How to Interpret Results

A photograph of a silver laptop on a dark surface. A semi-transparent blue hexagonal overlay is positioned over the laptop screen, showing a virtual desktop environment with a news website. The website has a "NEWS" header and several articles with images. The background is a dark, textured wall.

HOW TO USE IOMETER TO SIMULATE A DESKTOP WORKLOAD

Jim Moyle

To deliver VDI Performance it is key to understand I/O performance when creating your VDI architecture. Iometer is treated as the industry standard tool when you want to test load upon a storage subsystem.

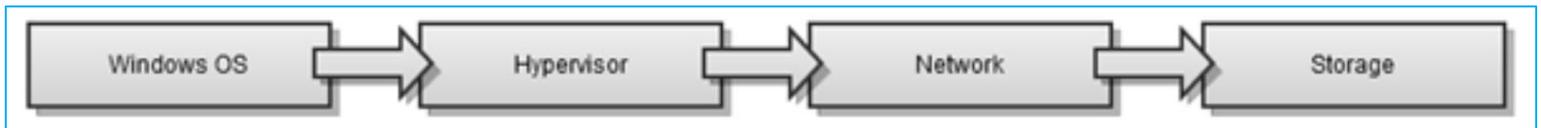
While there are many tools available, lometer's balance between usability and function sets it out. However, lometer has its quirks and this blog entry will attempt to show exactly how you should use lometer to get the best results, especially when testing for VDI environments. We will also show you how to stop people using lometer to fool you.

As lometer requires almost no infrastructure, you can use it to very quickly determine the storage subsystem performance. In steady state a desktop (VDI or RDS) I/O profile will be approximately 80/20 write/read, 80/20 random/sequential and the block size of the reads and writes will be in 4k blocks. The block size in a real windows workload does vary between 512B and 1MB, but the vast majority will be at 4K, as lometer does not allow a mixed block size during testing we will use a constant 4K.

That said, while lometer is great for analysing storage subsystem performance, if you need to simulate a real world workload for your VDI environment I would recommend using tools such as those from Login VSI.

Bottlenecks for Performance in VDI

lometer is usually run from within a windows guest which is sitting upon the storage subsystem. This means that there are many layers between it and the storage as we see below:



If we are to test the performance of the storage, the storage must be the bottleneck. This means there must be sufficient resource in all the other layers to handle the traffic.

Impact of Provisioning Technologies on Storage Performance

If your VM is provisioned using Citrix Provisioning Services (PVS), Citrix Machine Creation Services (MCS) or VMware View Linked Clones, you will be bottlenecked by the provisioning technology. If you test with lometer against the C: drive of a provisioned VMs you will not get full insight of the storage performance as these three technologies fundamentally change the way I/O is treated.

You cannot drive maximum IOPS from a single VM, it is therefore not recommended to run lometer against these VMs when attempting to stress-test storage.

I would always add a second drive to the VM and test lometer against a second hard drive as this by-passes the issue with PVS/MCS/Linked Clones.

In 99% of cases I would actually rather test against a 'vanilla' Windows 7 VM. By this I mean a new VM installed from scratch, without it joining the domain and only having the appropriate hypervisor tools installed. Remember, lometer is designed to test storage. By testing with a 'vanilla' VM environment you baseline core performance delivery. From that you can go to test a fully configured VM; and now you can understand the impact of AV filter drivers, provisioned by linked clones, or other software/agents etc. has on storage performance.

Using lometer for VDI testing: advantages and disadvantages

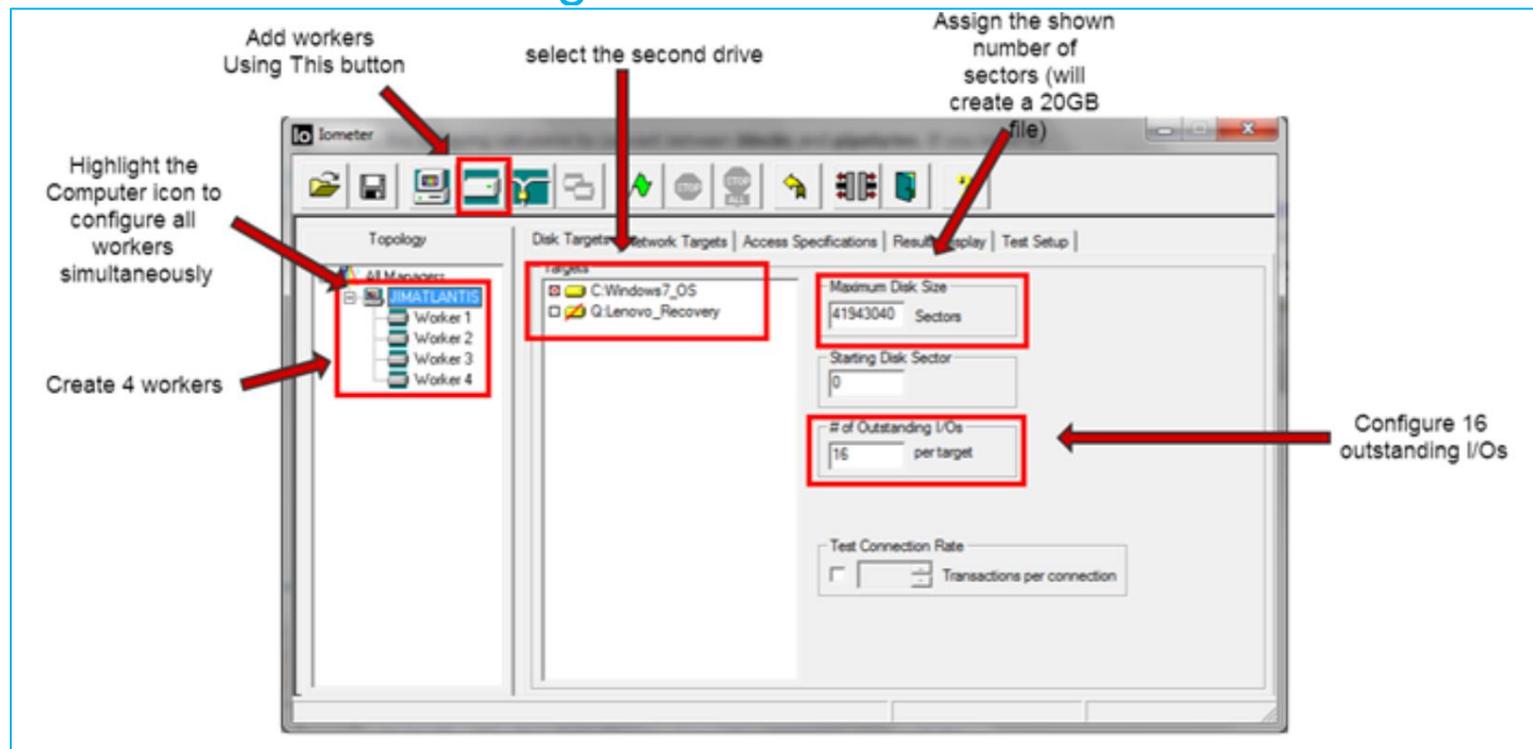
Before we move on to the actual configuration setting within lometer, I want to talk a little bit about the test file that lometer creates to throw I/O against. This file is called iobw.tst and is why I both love and hate lometer. It's the source of lometers biggest bugs and also it's biggest advantage.

First, the advantage; lometer can create any size of test file you like in order to represent the test scenario that you need. When we talk about a single host with 100 Win 7 VMs, or 8 RDS VMs, the size of the I/O 'working set' must be, at a minimum, the aggregate size of the pagefiles: as this is will be the a set of unique data that will consistently be used. So for the 100 Win 7 VMs, with 1GB RAM, this test file will be at least 100GB and for the 8 RDS VMs, with 10GB RAM, it would be at least 80GB. The actual working set of data will probably be much higher than this, but I'm happy to recommend this as a minimum. This means that it would be very hard for a storage array or RAID card to hold the working set in cache. lometer allows us to set the test file to a size that will mimic such a working set. In practice, I've found that a 20GB test file is sufficient to accurately mimic a single host VDI load. If you are still getting unexpected results from your storage, I'd try and increase the size of this test file.

Second, the disadvantage; iobw.tst is buggy. If you resize the file without deleting, it fails to resize (without error) and if you delete the file without closing lometer, lometer crashes. In addition, if you do not run lometer as administrator, Windows 7 will put the iobw.tst file in the profile instead of the root of C:. OK, that's not technically lometer's fault, but it's still annoying.



Recommended Configuration of Iometer for VDI workloads



First tab (Disk Targets)

The number of workers is essentially the number of threads used to create the I/O requests, adding workers will add latency, it will also add a small amount of total I/O. I consider 4 workers to be the best balance between latency and IOPS.

Highlighting the computer icon means that all workers are configured simultaneously, you can check that the workers are configured correctly by highlighting the individual workers.

The second drive should be used to avoid issues with filter drivers/provisioning etc on C: (although Iometer should always be run in a 'vanilla' installation).

The number of sectors gives you the size of the test file, this is extremely important as is mentioned above. You can use the following website to determine the sectors/GB:

<https://www.unitconversion.org/data-storage/blocks-to-gigabytes-conversion.html>

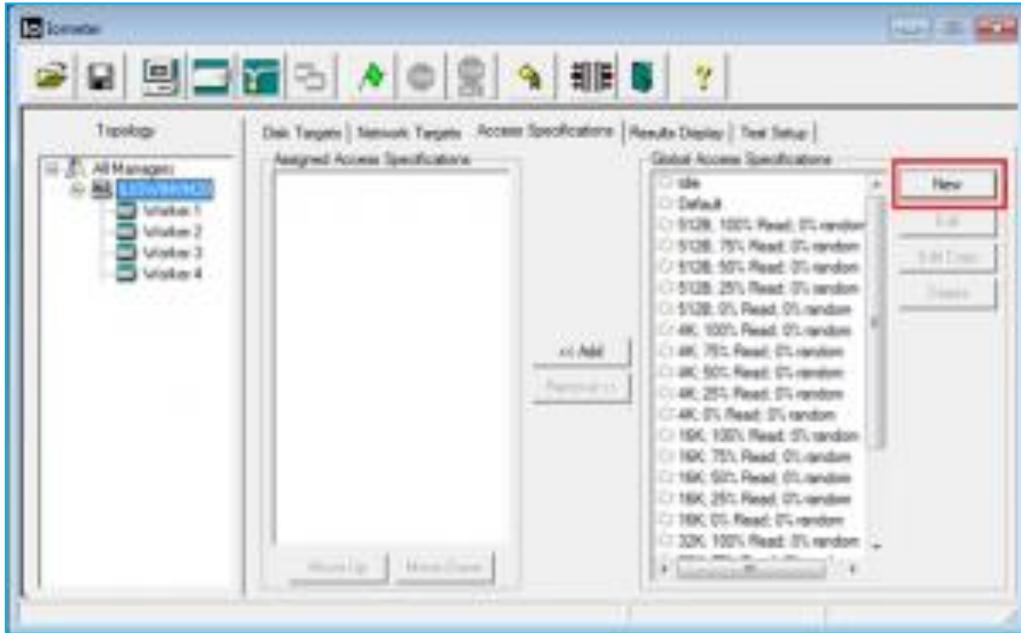
The size used in the example to get 20GB is 41943040 sectors.

The reason for configuring 16 outstanding I/Os is similar to the number of workers as increasing I/Os will increase Latency while slightly increasing IOPS. As with workers, I think 16 is a good compromise. You can also refer to the following article regarding outstanding I/Os: <https://communities.vmware.com/docs/DOC-3961>

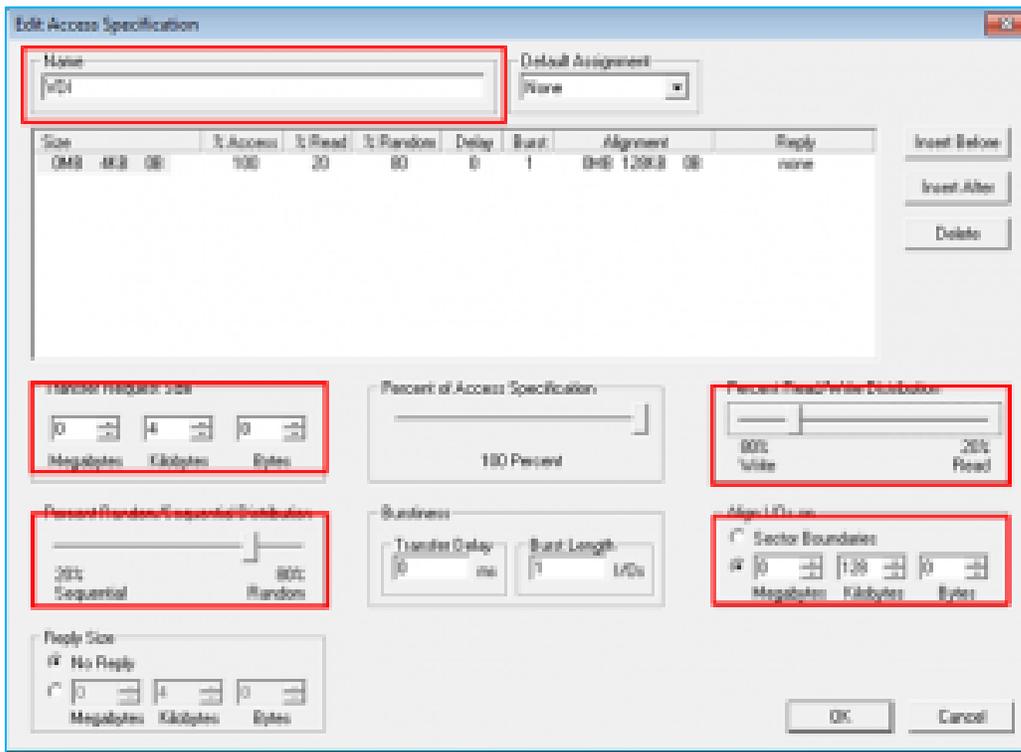
Second tab (Network Targets)

No changes are needed on the network Targets tab

Third tab (Access Specifications)

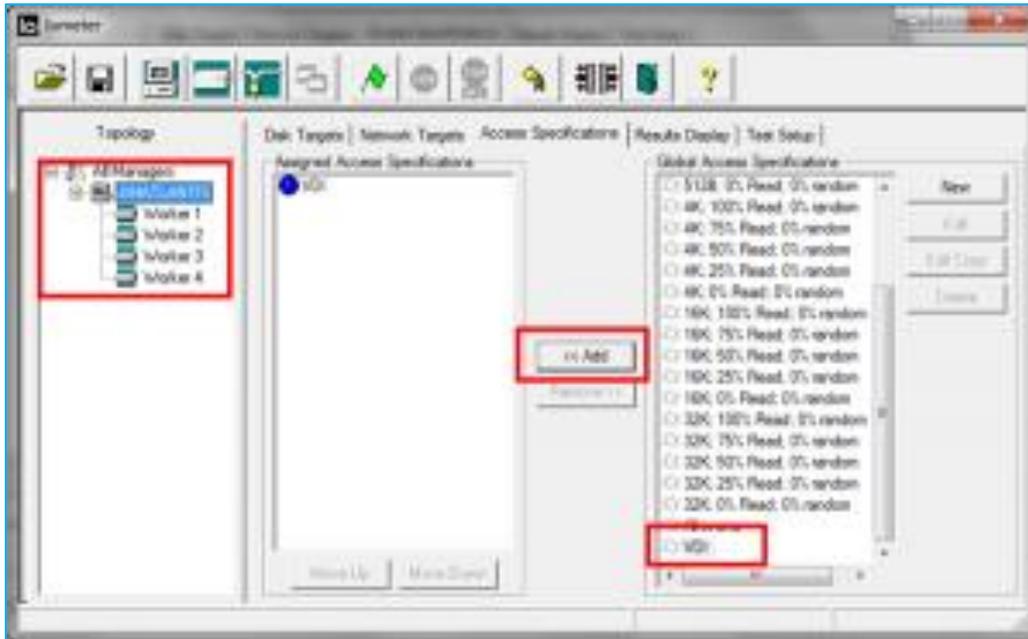


To configure a workload that mimics a desktop, we need to create a new specification.



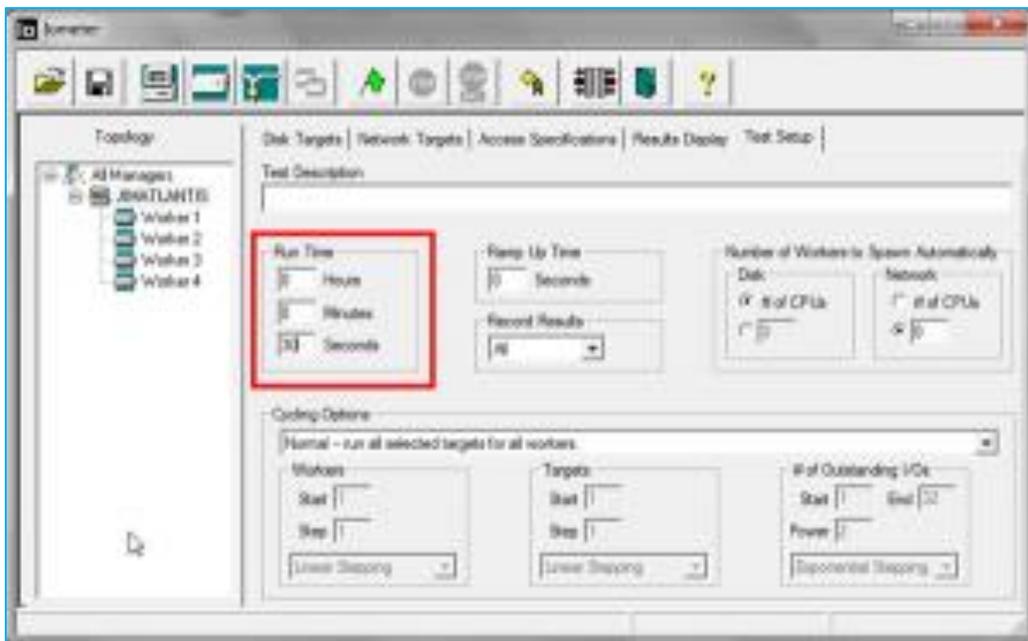
The new Access specification should have the following settings. This is to ensure that the tests model as closely as possible a VDI workload. The settings are:

- 80% Write
- 80% Random
- 4K blocks
- Sector Boundaries at 128K, this is probably overkill and 4K would be fine, but should eliminate any disk alignment issues.



You should then Add the access specification to the manager.

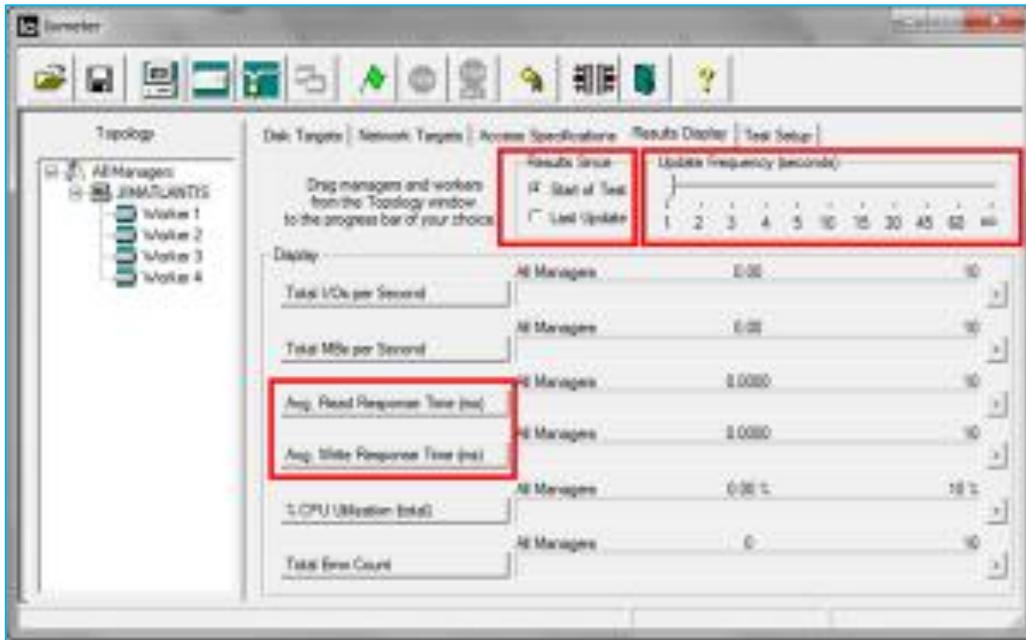
Fifth tab (Test Setup)



I'd advise only configuring the test to run for 30 seconds, the results should be representative after that amount of time. More importantly, if you are testing your production SAN, Iometer once configured correctly

will eat all of your SAN performance. Therefore, if you have other workloads on your SAN, running Iometer for a long time will severely impact them.

Fourth tab (Results Display)



Set the Update Frequency (seconds) slider to the left so you can see the results as they happen.

Set the 'Results Since' to 'Start of Test' which will give you a reliable average.

Both Read and Write avg. response times (Latency) are essential.

It should be noted that the csv file Iometer creates will capture all metrics while the GUI will only show six.

Save Configuration

It is recommended that you save the configuration for later use by clicking the disk icon. This will save you having to re-configure Iometer each test run you do. The file is saved as *.icf in a location of your choosing. Or to save some time, download a preconfigured Iometer Desktop Virtualization Configuration file and load it into Iometer.

Start the test using the green flag.

Interpreting Results

Generally, the higher the IOPs the better, indicated by 'total IOPS per second' counter above, but this must be delivered at a reasonable latency, anything under 5ms will provide a good user experience.

Given the max possible IOPS for a single spindle is 200, you should sanity check your results against predicted values. For an SSD you can get 3-15,000 IOPS depending on how empty it is and how expensive it is, so again you can sanity check your results.

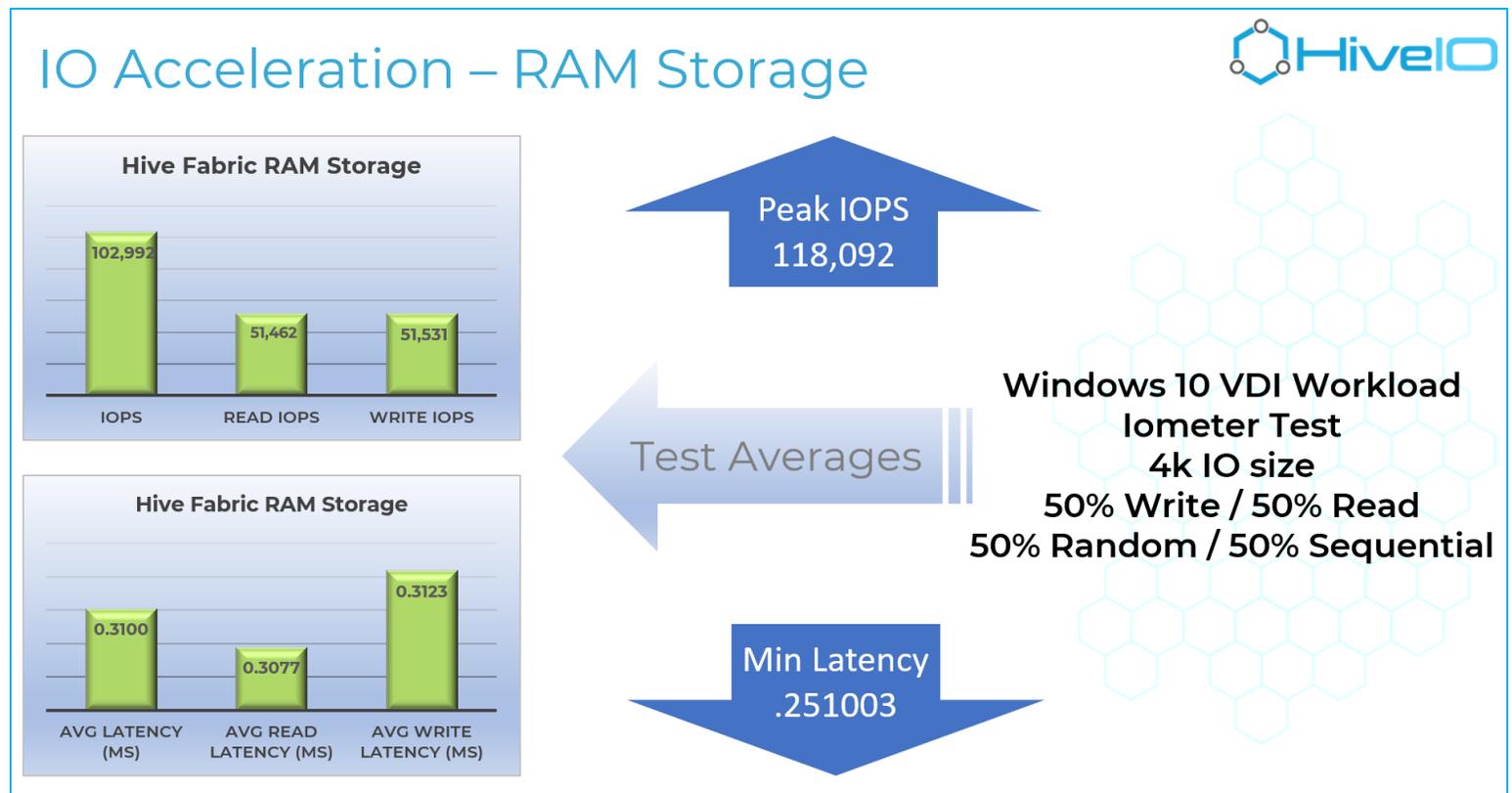
You don't need to split IOPS or throughput out into read and write because we know Iometer will be working at 80/20, as we configured in the access specification.

How can I check someone isn't using Iometer to trick me?

To the untrained eye Iometer can be used to show very unrepresentative results. Here is a list of things to check when someone is showing you an Iometer result.

- What size is the test file in Explorer? it needs to be very large (minimum 20GB), don't check in the Iometer gui.
- How sequential is the workload? The more sequential, the easier it is to show better IOPS and throughput. (It should be set to a minimum of 75% random)
- What's the block size? Windows has a block size of 4K, anything else is not a relevant test and probably helps out the vendor.

Iometer Results with Hive Fabric For VDI



Try Hive Fabric Today - <https://www.hiveio.com/demo/>