Windows 7 IOPS for VDI: Deep Dive

Jim Moyle

Lead Northern Europe Solutions Consultant

Atlantis Computing

Contents

Summary

Windows 7 and IOPS is the most challenging, least understood and most frequently encountered resource bottleneck that stalls todays desktop virtualization projects. Many of these projects are Windows 7 migrations which are one of the key IT initiatives which are driving VDI adoption today. However, many IT organizations planning a Windows 7 migration with VDI are unaware that the combination of Windows 7 and Anti-virus more than *doubles* the amount of memory and IOPS required per desktop compared to Windows XP, which will significantly decrease virtual desktop density per server and will degrade virtual desktop performance. As a result, the storage architecture is often undersized for Windows 7. This leads to desktop performance and budgeting issues when more storage is required to fix the IOPS bottleneck.

In the document below I have tested the IOPS resource requirements for Windows 7 in different situations giving the enterprise architect the information needed to correctly size storage for Virtual Desktop projects.

Why IOPS is Key

Delivering a good desktop experience, whether it is a physical PC or a virtual desktop, is a matter of ensuring that it has sufficient hardware resources (e.g. CPU, Memory, Storage) to run the operating system and applications of the desktop. With Physical PCs, the process is simple because each of those resources is both local to the PC and dedicated for that PC. In desktop virtualization, we are abstracting and pooling those hardware resources. With memory and CPU the pool of resources is limited to the hypervisor and the hence the physical server, which remains local to the desktop and is easily predictable and handled well by a variety of memory and CPU optimisation techniques. However, with desktop virtualization, the hard drive is moved to shared storage, this storage is then shared between hypervisors, allowing consequences to spread across the whole infrastructure. In order to deliver a consistently high performance virtual desktop experience that is equal or better than a physical PC, virtual desktops require constant access to low latency and high throughput storage.

IOPS and Scaling VDI for Production

Many IT organizations have delivered successful VDI pilot projects, only to fail as they scale the production rollout. During the pilot, the shared storage infrastructure delivers more than adequate throughput, providing excellent desktop performance. However, as the pilot transitions to production, storage infrastructure IO performance degrades as each virtual desktop makes requests of the shared storage infrastructure as if it were dedicated storage. The result is poor desktop and application performance and increasing user dissatisfaction with the virtual desktop infrastructure. With conventional VDI environments, the only way to address this degraded performance is to spread the virtual desktop load over more and more drives and storage controllers. This either increases the cost of VDI beyond the original budget or dooms the project to failure.

Desktop Delivery model: Stateless or Persistent?

Before we talk about VDI, it will be useful to move back in time slightly to Microsoft Terminal Server, (now known as Session Based RDS) . Session based RDS Solutions can also give us a virtual desktop with a remoting protocol, giving us many of the benefits of VDI solutions.

A major reason that session based solutions never had the widespread adoption that many of us would have liked was the engineering cost, whether that is the cost of repackaging applications, engineering a locked down OS, or providing a new profile and printing solution. These costs limited Terminal Services to specific use cases and industry sectors.

The promise of VDI was to provide a desktop over remote protocol experience which doesn't have some of the issues of a Terminal Server deployment. This basically means a VDI solution should be able to be implemented and maintained with a fraction of the engineering cost of a session based solution.

This brings us right back to the deployment model, if we split up the application, user and OS part of the VDI stack, we can get to the stateless model which is touted as the best practice by the Desktop Virtualisation vendors.

The thing is, creating a stateless environment is a complex undertaking. Repackaging all applications for virtualisation (streaming), virtualising the user and optimising and locking down the OS puts us right back into that bracket of high engineering time for supporting and implementing a virtual desktop. At which point, you might as well use Session based RDS as it has a lower CAPEX.

When you look at the massive effort and cost in desktop transformation required to go stateless, the cost/benefit equation of VDI doesn't look so good to a CEO. I'm not alone in thinking this and I see a great many customers eschewing the stateless model and opting for persistent desktops. Not only persistent desktops, but an unmodified Windows desktop that requires little change from their traditional Windows physical PC image as possible. Obviously these persistent VMs have a much larger hardware requirement. However, for the largest VDI projects, I believe that the trend is that IT organizations are first starting with persistent VDI to avoid the upfront engineering effort to go stateless with perhaps having a goal of later converting to a stateless model over time. In fact 8 out of the 9 largest VDI project (10,000-100,000 desktops) that I know of in the financial sector are starting with the persistent VDI model.

In general, most of the people implementing VDI will be in a state somewhere in-between these extremes, whether due to design considerations, cost of implementation, or timeline pressure. We should at least know the consequences of our decisions and how to frame our advice in a meaningful way before blindly recommending stateless VDI designs based on its engineering elegance without taking into account the amount of change required by the IT organisation.

Competing for shared resources

Windows 7 was designed with a local and dedicated disk and requires constant access to the hard drive even when it is idle. In addition, Windows 7 will consume as much disk IO or throughput to the hard drive as is possible. This means that Windows isn't a good community member, it is very selfish. As a result, when you virtualize thousands of Windows 7 desktops and have them share storage (SAN or NAS), they all compete to use as much disk IO as possible to maximize their own performance without any knowledge of the needs of the other desktops that are sharing that same hardware resource. If a Windows 7 desktop doesn't have access to the IO that it requires, desktop performance will degrade and users will see it in the form of long boot/logon times, slow application launching and generally poor desktop performance. Also, organizations will be limited in how they can boot their virtual desktops, when and how they can patch the operating system and applications and when they can run Anti-Virus scans.

Windows 7 Queuing

So how does Windows manage this desire to eat resources? It basically introduces queuing: it will back up the read and write requests into queues and service them as resource becomes available, but, queuing introduces latency and this latency introduces a performance hit. This issue gets worse when we introduce virtualisation and shared storage as each storage layer introduces more queuing technologies and hence more latency. Storage latency is one of the key factors for Windows performance. Going from <2ms for a local drive to >25ms for a virtual infrastructure when contention is involved.

Microsoft Windows optimisations

Optimising Windows is a good thing, but how far can it get you? With maximum tuning, it is possible to reduce steady state IO requirement by 10-25%. This is a great result, but can come at the cost of some functionality, for instance disabling Windows indexing can hamper searching. While 10-25% may seem like quite a lot, as you will see below, such a gain in steady state IOPS may not be significant. I'd like to look at three Windows I/O optimisations that I commonly see recommended when doing VDI.

SuperFetch

Microsoft was so proud of this technology it trademarked the name. It was introduced in Windows Vista and attempts to predict the user behaviour. It records the user behaviour and tries to load the right pages into memory before we even know we need them. Pretty clever, right? Well, not really. Microsoft calls this a performance feature not an IOPS optimisation feature, the reason being is that it actually hurts disk usage by trying to predict the user behaviour and reading information from disk. If it is at all wrong, you have basically increased your IOPS.

I'd advise turning it off for VDI by disabling the service.

ReadyBoost

ReadyBoost is a USB Cache that means windows can use a fast flash drive as a read cache if it is inserted into a desktop/laptop. This functionality not relevant for VDI workloads.

ReadyDrive

Ready Drive enables Windows to take advantage of specialist hardware, Hard disk drives which have on-board flash memory. This again, is not relevant for VDI workloads.

So why is IOPS usually the limiting factor with VDI?

Moores law is basically responsible, meaning that CPU development has outstripped both memory and storage, especially storage.

Essentially the problem comes down to the physics of a spinning disk. In a traditional HDD there is a spinning platter which is why all disks have an RPM value. Although we can have faster disks, the expense in power, heat, noise and materials currently mean we can't effectively get faster commercial disks.

Each disk can provide 65-200 IOPS per spindle depending on its type (SATA, SAS). This figure cannot be increased at the moment. Essentially, to cover the amount of IOPS required by VDI by using traditional methods, huge amounts of HDDs and controllers need to be purchased.

With the emergence of Solid State Drives (SSDs), many organizations have tried to apply this technology to VDI in several different ways:

1. **As a Caching Tier on SAN/NAS**
2. **By creating Hybrid SSD/SAS storage arrays**
3. **As local disk to store the Base Image as a read-cache**

Using SSDs as local disks limits the deployment model in almost all cases to Stateless, this is because a persistent deployment means that users have essential data within the OS, whether that data is user installed applications, customisations, or actual data. If you only use local storage, it means that vMotion, FT and DRS are not available to protect the virtual machines, the absence of this protection causes most enterprises to discount local storage for persistent virtual desktops. SSDs are also expensive compared to normal drives, often don't work with blade servers as you can't fit enough drives in to give you the needed capacity.

Virtualisation is meant to save us space, power and cooling in the datacentre. If you have to add racks of HDDs to the datacentre, you essentially lose the benefits of virtualisation. This means your CAPEX/OPEX costs and the complexity of the VDI deployment goes up considerably.

What is the advice?

Various vendors both of VDI solutions and storage solutions will give varying answers when you ask them what IOPS you need per desktop in the world of VDI.

Here is the official word from Citrix:

Group	Operating System	vCPU Allocation	Memory Allocation	Avg IOPS (Steady State)	Estimate Users/Core
Light	Windows XP	1	768MB-1 GB	3-5	10-12
	Windows 7	1	1-1.5 GB	4-6	8-10
Normal	Windows XP	1	1-1.5 GB	6-10	8-10
	Windows 7	1	1.5-2 GB	8-12	6-8
Power	Windows XP	1	1.5-2 GB	12-16	6-8
	Windows 7	1-2	2-3 GB	15-25	4-6
Heavy	Windows XP	1	2 GB	20-40	4-6
	Windows 7	2	4 GB	25-50	2-4

Other virtualization and storage vendors have similar views on IOPS. Yet the average steady state IOPS is of little relevance when designing a Virtual Desktop Infrastructure. To give best experience, you need to accommodate the heaviest activity.

Measuring IOPS

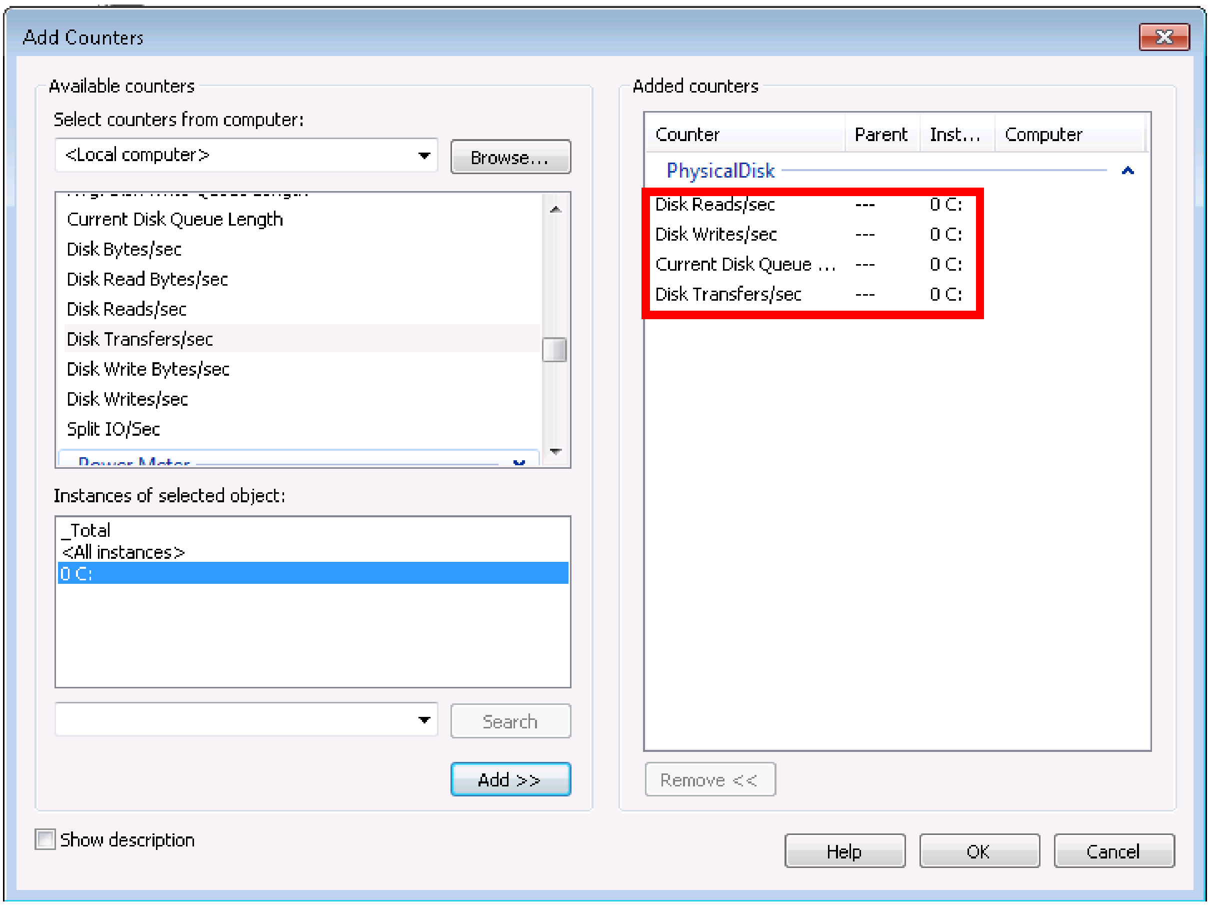
Measuring IOPS can be a confusing issue as it does not correlate directly to performance. It is a combination of IOPS requested and IOPS available that will give us storage latency. Storage latency has a direct correlation to performance as a large latency results in the OS and applications becoming unresponsive.

We'd like to be able to measure the IOPS requested by Windows, but there is no direct way to do this, there is only the ability to measure the IOPS delivered. By eliminating the queues as much as possible and trying to create an environment where Windows has all the IOPS it needs, we hope to achieve a state where the Windows IOPS requested is as close to the IOPS delivered as possible.

How to measure IOPS delivered

IOPS delivered can be measured in a few ways; one most familiar to us is performance monitor, which can give us some very interesting disk performance statistics:

Perfmon



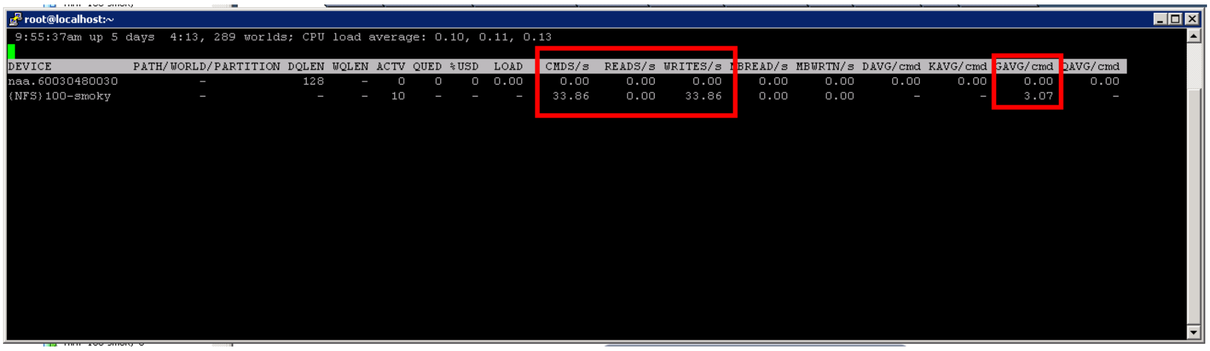
Performance Monitor Statistics

- Disk reads/sec is read IOPS
- Disk Writes per/sec is Write IOPS
- Disk Transfers/sec is total IOPS
- Current Disk queue length will give us how many IOPS are backed up in Windows

The current disk queue length gives us an approximation of Latency. The perfmon counters can be easily exported and graphed for study.

ESXTOP

Also we can look at ESXTOP, you need to SSH into ESX and run the command ESXTOP, which will provide disk stats. It is also advisable to change the display to update every two seconds, by hitting s 2.



The two highlighted areas above are the ones we are particularly interested in:

1. CMDS/s is total IOPS with the equivalent Write and read highlighted to the right.
2. On the far right, we have GAVG/cmd, which will give us the latency.

ESXTOP counters can also be exported and graphed.

XenServer also has monitoring capabilities, but they are not covered in this document.

For Hyper-V, perfmon can be used to perform the same function.

Testing your Infrastructure

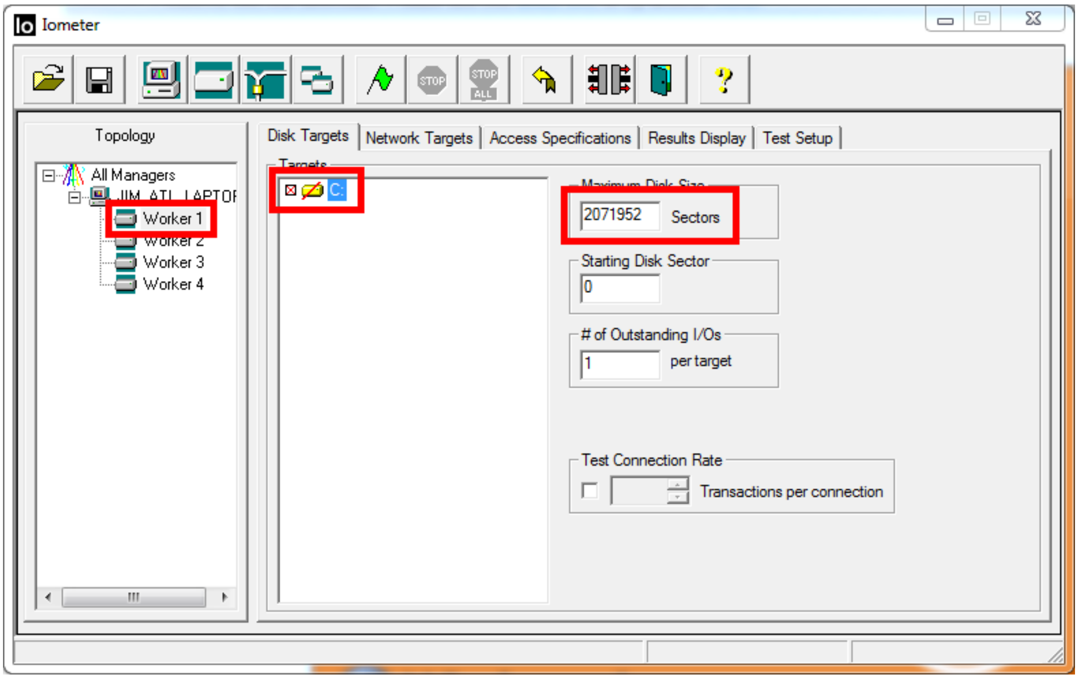
It is worth spending a little time talking about how to test your infrastructure once you have it stood up, it is essential to test to check that everything is working correctly. The results of your tests can also ensure that your tuning has had the desired effect. There are a number of tools to do this, notably IOMeter and WINSAT to measure maximum IO and Login VSI to measure VM density and user experience.

IOMeter

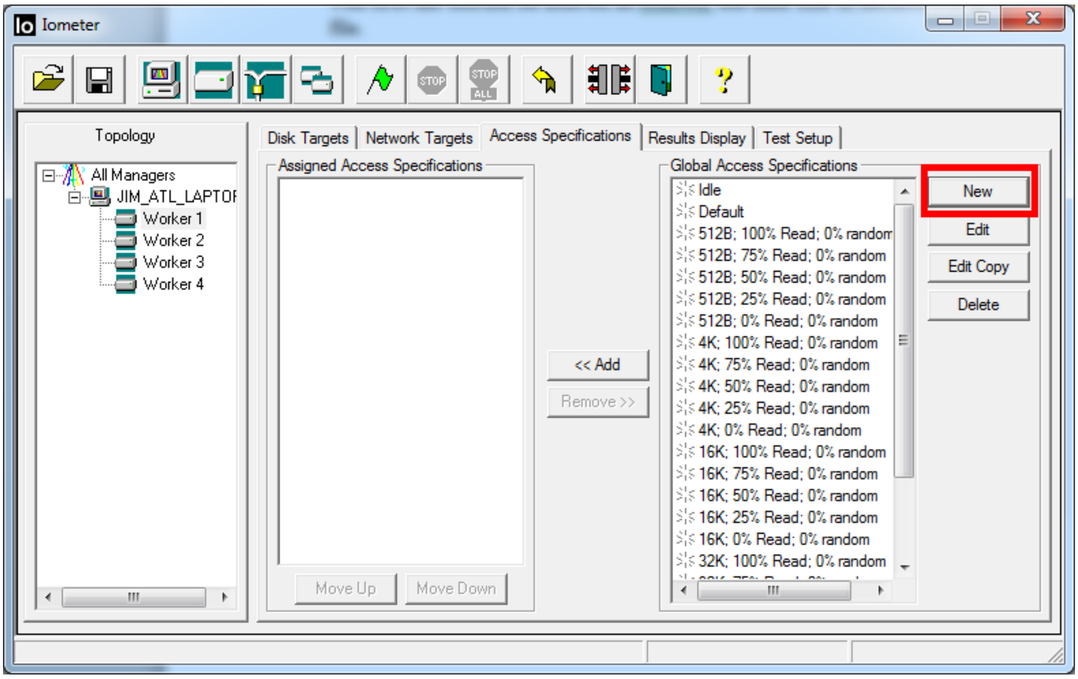
IOMeter is an open source tool to measure peak IOPS with a certain workload. Although IOMeter is a little buggy and can report inaccurately if you are not used to its foibles, it is the best publically available tool that I know of to measure max IO.

Here is how I configure IOMeter to measure the max IO for a workload that is as close as possible to a VDI IO profile.

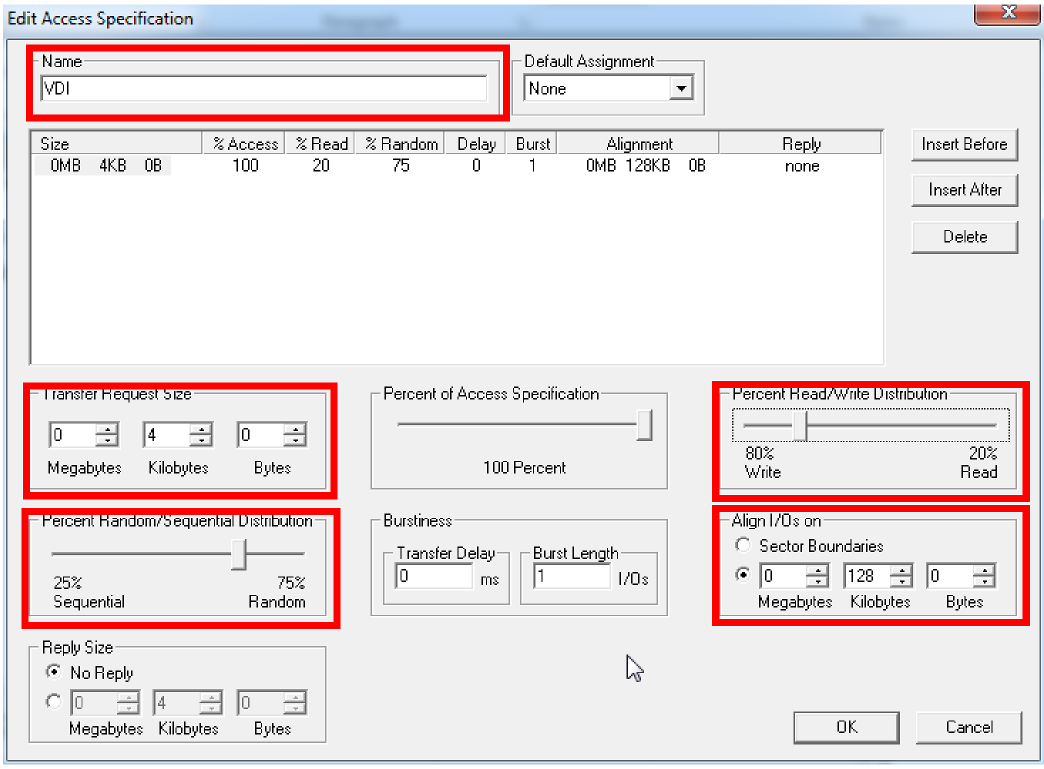
The first tab should be altered as below with the disk size in sectors that will use a 2GB test file.



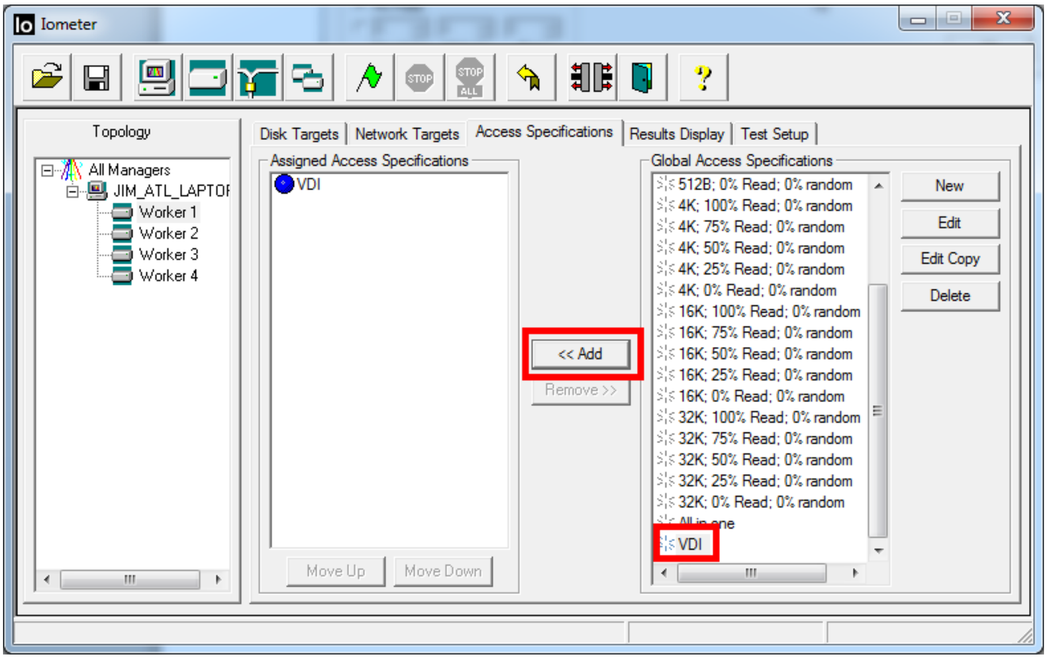
A new workload should be generated.



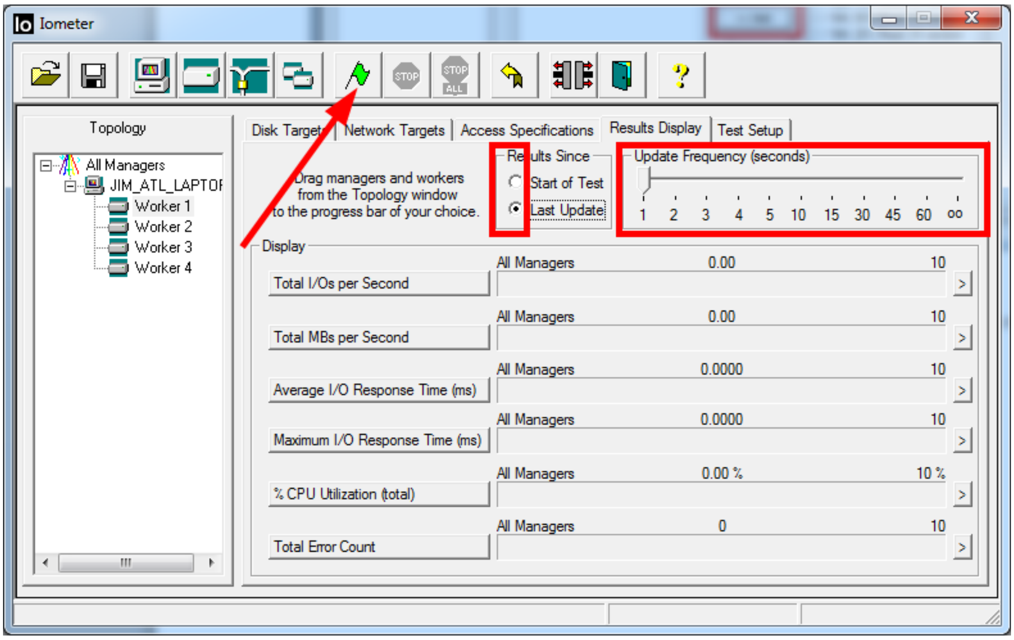
The new Access specification should have the following settings. This is to ensure that the tests model as closely as possible a VDI workload.



You should then add the access specification to the worker.



The test should then be started by pressing the green flag, to show the results in real time, move the slider all the way to the left.



The 'total I/Os per Second' is the IOPS available to that VM

Windows System Assessment Tool

WinSAT is a command line version of the GUI system assessment tool, which gives your PC its rating.

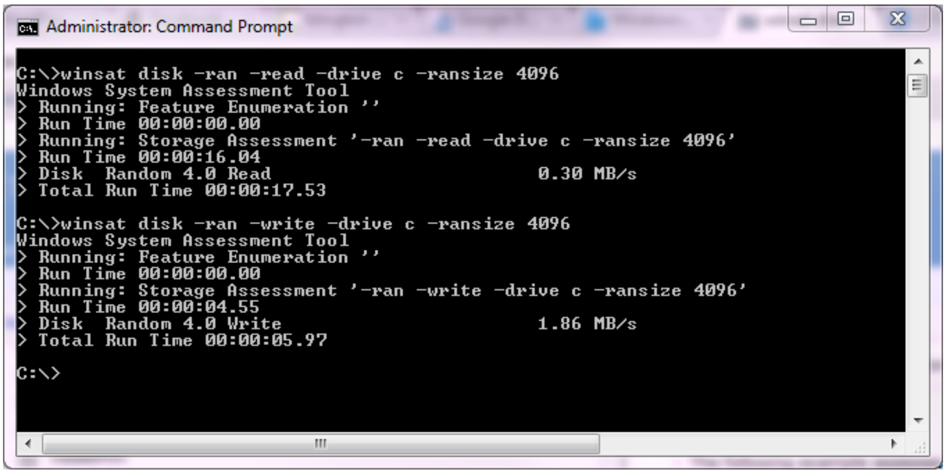
The WinSAT tool will also analyse the IO capabilities of a disk, although it is a much blunter tool than IOMeter and should only be used when IOMeter is not available. It can only measure read or write and not a combination of both.

The two command lines I recommend are:

winsat disk -ran -read -drive c -ransize 4096

winsat disk -ran –write -drive c -ransize 4096

The resulting display will look as below:



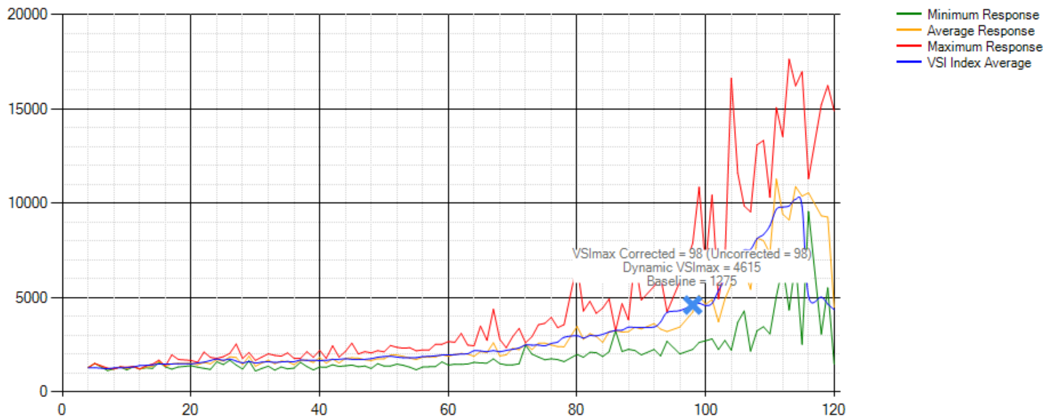
Login VSI

Login VSI is now considered the industry standard to create real world workload generation on VDI infrastuctures. It can be obtained from the Login Consultants site. It is the only tool available that will accurately simulate real world users on your infrastructure and then report in a meaningful way.

a. The login VSI tool takes a little while to setup and will need the appropiate software in the Win7 master image.

b. The tool will measure the maximum number of VMs that the tested infrastructure can handle before the user experience degrades beyond the point of being productive.

Below is an example of the chart that the Login VSI tools will create.



The VSI Max score is the number of desktops that the infrastructure can handle before the user experience degrades.

ESX memory management

Although slightly off topic for this discussion, ESX memory management deserves a mention. ESX memory management can become a large source of IOPS, as when it is under pressure it basically swaps Memory for either CPU or IOPS.

Transparent page sharing swaps Memory for CPU and doesn't concern us here.

Memory Ballooning forces the guest VM to operate with less memory by reserving an amount of assigned memory. The big problem with this is that ESX has no clue how much of the guest memory is being used. It basically removes memory from the VM and hopes it won't lead to swapping. Ballooning will kick in before the ESX host has run out of memory, at about 5% free.

Hypervisor swapping is the creation of the infamous vSwap file. This is basically paging at the hypervisor level. It means that when the ESX host comes under severe memory pressure it will use the swap file instead of physical memory.

The combination of Ballooning and Hypervisor swapping can lead to a 'double page' of memory where the information is swapped to disk twice.

This means memory overcommit in ESX VDI environment should be used with extreme caution, especially in environments where you are already limited by IOPS.

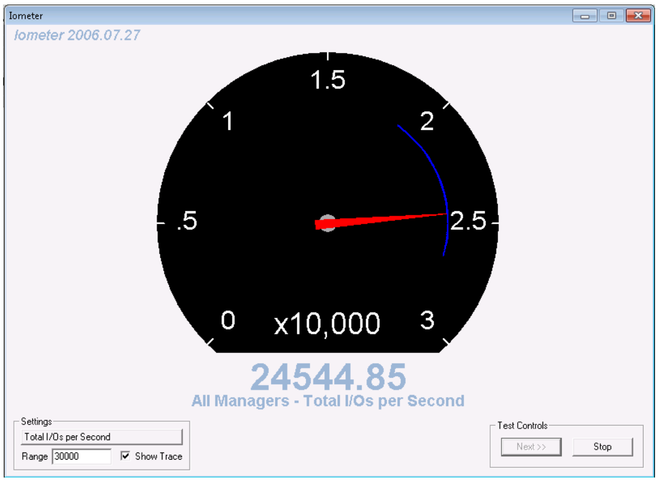
Determining Windows demand for IOPS

To test my assertion that Windows 7 will consume as many IOPS it has access to, I wanted to look at what a Windows 7 VM IOPS profile looked like when I removed the IOPS limit. Basically I'd try and take out the influence of queuing and the latency that ensues in a 'normal' environment. This means that I can see the real demand that Windows has for IOPS. As I remove the IOPS limit, I will also observe the effect on performance. My hypothesis is that the number of IOPS required per Windows 7 desktop is not a simple number based on the type of user and operating system but includes many more factors such as the desired performance during boot/logon/logoff, steady state operations, patches, anti-virus scans and other frequent activities. As we decide how many IOPS to design for our VDI environments, we should try to deliver a virtual desktop experience that is better than a physical PC and at least be aware and understand the consequence of limited IOPS below that bar.

Test Configuration

The testing was done with a single, unmodified Windows 7 image, with 2 GB RAM and 2 vCPUs running against a Linux RAM drive backend.

The storage backend had approximately 23,000 IOPS available as seen below. The data was collected using a variety of methods including ESXTOP, Perfmon and the Linux Dstat tool.



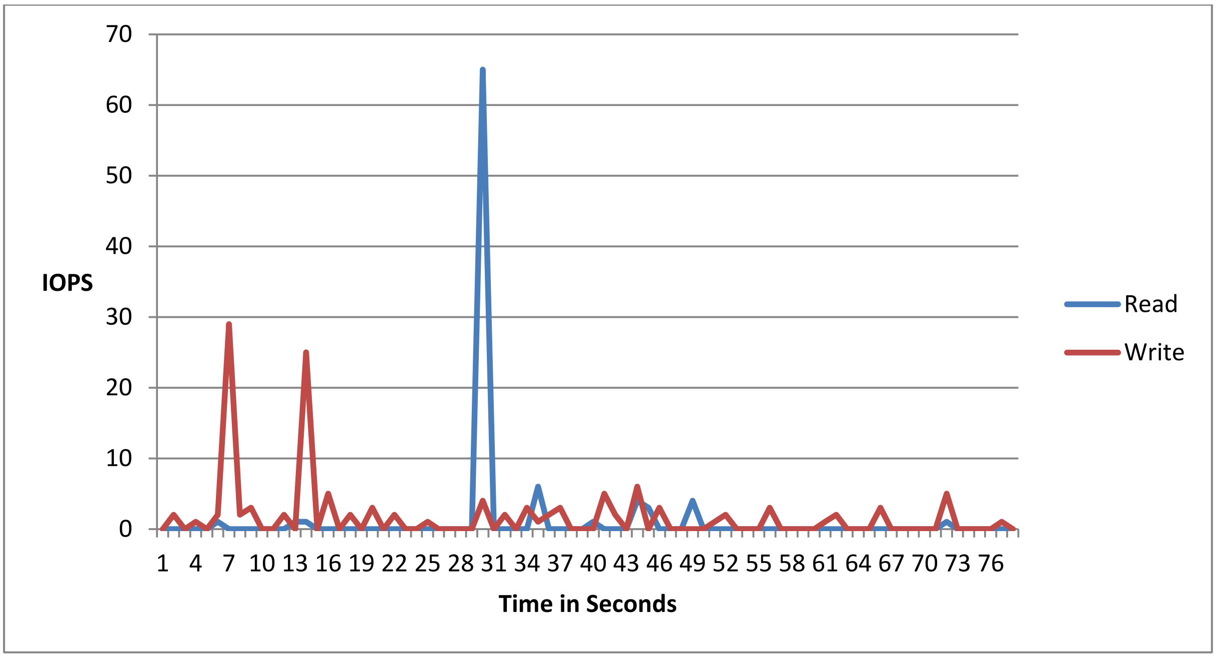
To simulate load on the system, I used the Login VSI tool from Login consultants.

Test Results

The results from various test situations are shown below, the format for all graphs is the same no matter how the data was collected.

Windows post boot, pre logon

First we can look at a graph where Windows is sitting there doing nothing before logon.

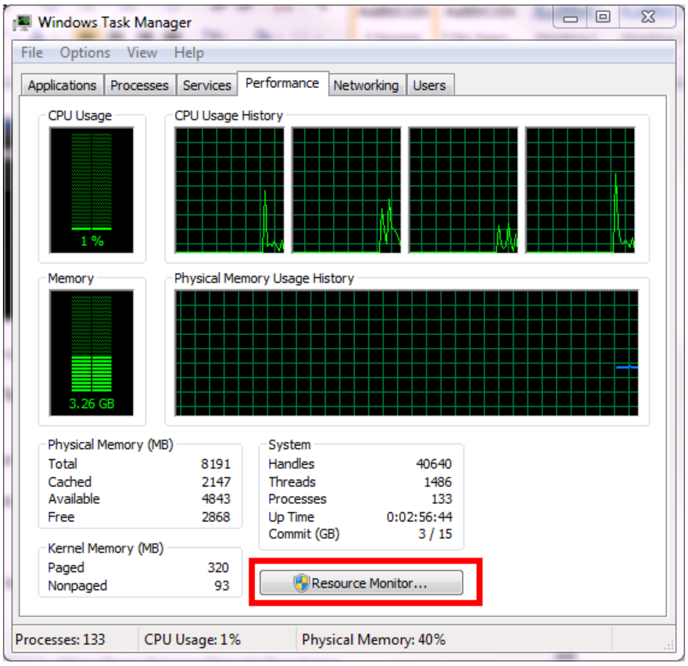


As we can see we have a peak of around 65 IOPS for read and a smaller peak, but higher average for write, already we are looking at a graph where we have a low average and a large peak.

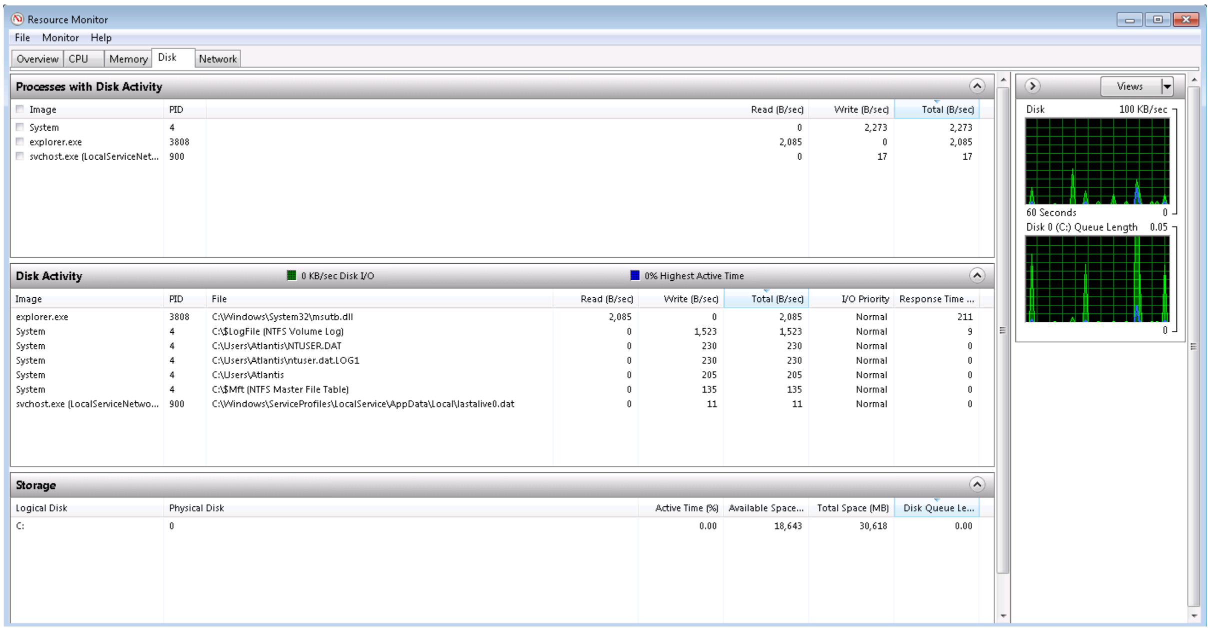
The averages are read: 0.94

Write: 1.4

So what is that peak? We need to introduce some more tools at this point. As part of the new task manager in Win7, we have resource monitor.



Once we have accessed resource monitor, we see the screen below.



Resource Monitor will show us the IO throughput in B/sec over the last 60 seconds. It is important to make the distinction between B/sec and IOPS. Throughput and IOPS can be generally related together as windows has a relatively consistent block size, though not always.

This is due to how Windows Memory works, memory pages are 4K in size, as such windows will load files into memory in 4K blocks, this means that most of the read and write activity has a 4K block size. Windows 7 does try and aggregate sequential writes to a larger block size to make writing a more efficient process. It will try and aggregate the writes to up to 1MB in size. The reason for this is that again Windows is expecting a local, dedicated spindle and spinning disks are very good at writing large blocks.

If you add to this that the I/O then has to go through ESX which further splits and randomises it, you end up with an I/O blender effect, making almost all blocks 4k and random. Disks are very bad at writing many small random blocks.

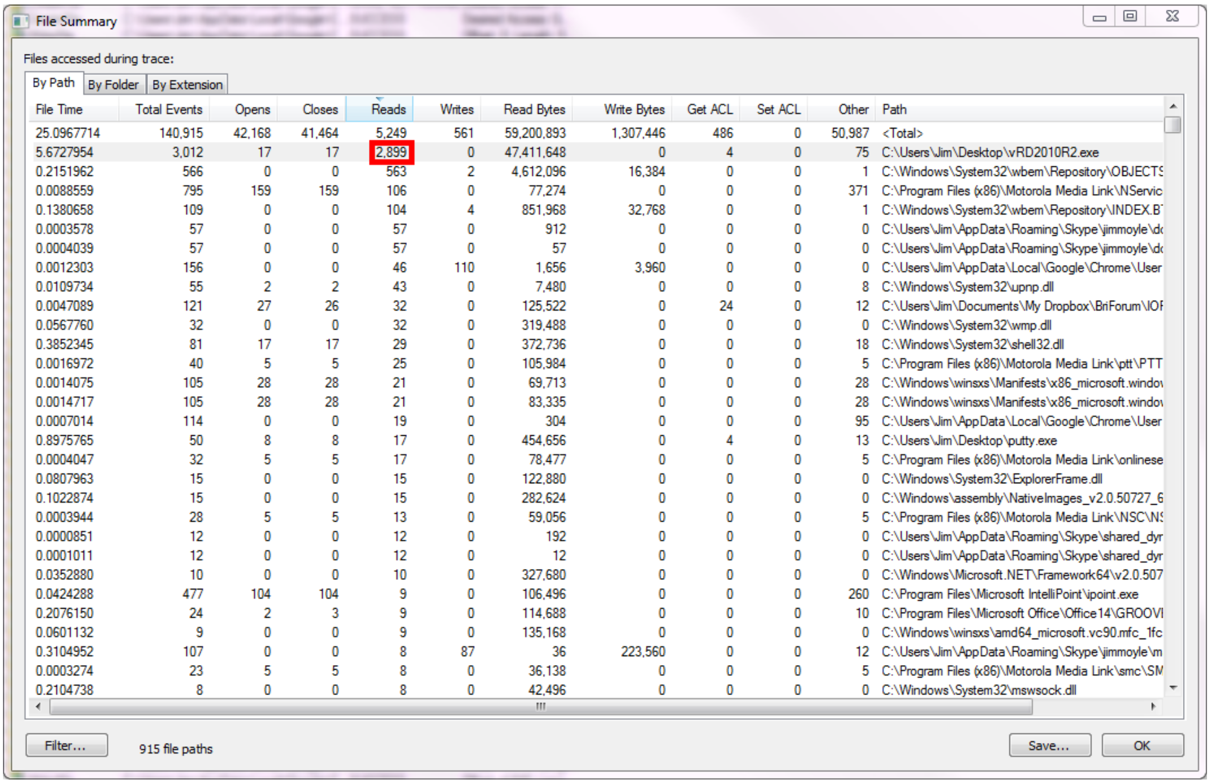
With some careful watching of resource monitor and perfmon, you can correctly identify the reason for the spikes.

If you wish to be able to optimise windows for VDI, eliminating the spikes is key as you need to be able to cope with peak activity.

There is a better tool available for analysing IOPS in windows which is Process Monitor from Sysinternals.

As you can see below, process monitor has a file summary view. This view reflects the currently filtered events. With a process trace and the time of the spike we can narrow down the culprit very effectively.

In the case below I have identified the cause of a large read spike as a remote desktop tool I'm using.


File Summary

Files accessed during trace:

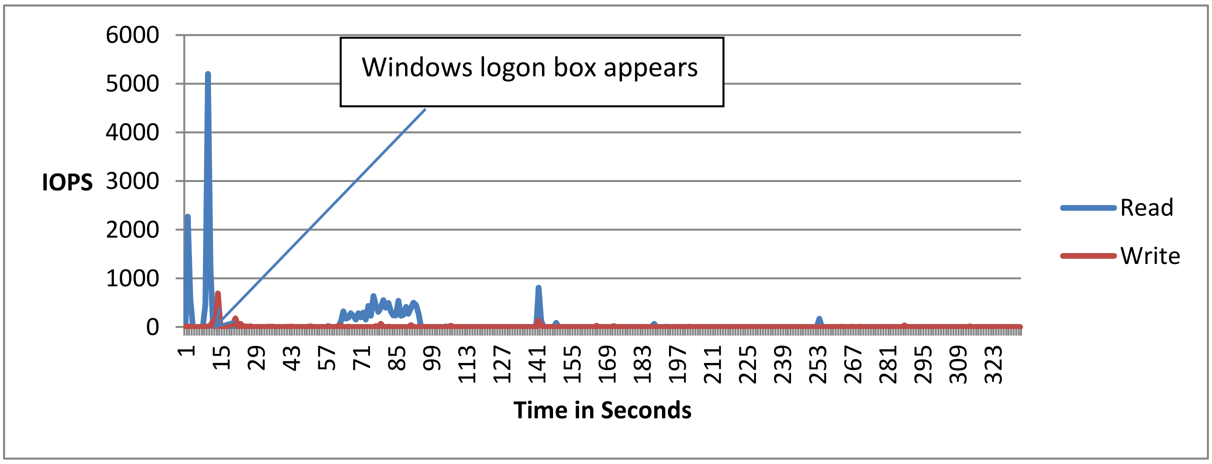
By Path | By Folder | By Extension

File Time	Total Events	Opens	Closes	Reads	Writes	Read Bytes	Write Bytes	Get ACL	Set ACL	Other	Path
25.0967714	140,915	42,168	41,464	5,249	561	59,200,893	1,307,446	486	0	50,987	<Total>
5.6727954	3,012	17	17	2,899	0	47,411,648	0	4	0	75	C:\Users\Jim\Desktop\vRD2010R2.exe
0.2151962	566	0	0	563	2	4,612,096	16,384	0	0	1	C:\Windows\System32\wbem\Repository\OBJECTS
0.0088559	795	159	159	106	0	77,274	0	0	0	371	C:\Program Files (x86)\Motorola Media Link\NServic
0.1380658	109	0	0	104	4	851,968	32,768	0	0	1	C:\Windows\System32\wbem\Repository\INDEX.B
0.0003578	57	0	0	57	0	912	0	0	0	0	C:\Users\Jim\AppData\Roaming\Skype\jimmoyle\d
0.0004039	57	0	0	57	0	57	0	0	0	0	C:\Users\Jim\AppData\Roaming\Skype\jimmoyle\d
0.0012303	156	0	0	46	110	1,656	3,960	0	0	0	C:\Users\Jim\AppData\Local\Google\Chrome\User
0.0109734	55	2	2	43	0	7,480	0	0	0	8	C:\Windows\System32\upnp.dll
0.0047089	121	27	26	32	0	125,522	0	24	0	12	C:\Users\Jim\Documents\My Dropbox\BriForum\IO
0.0567760	32	0	0	32	0	319,488	0	0	0	0	C:\Windows\System32\wmp.dll
0.3852345	81	17	17	29	0	372,736	0	0	0	18	C:\Windows\System32\shell32.dll
0.0016972	40	5	5	25	0	105,984	0	0	0	5	C:\Program Files (x86)\Motorola Media Link\ptt\PTT
0.0014075	105	28	28	21	0	69,713	0	0	0	28	C:\Windows\winsxs\Manifests\x86_microsoft.windo
0.0014717	105	28	28	21	0	83,335	0	0	0	28	C:\Windows\winsxs\Manifests\x86_microsoft.windo
0.0007014	114	0	0	19	0	304	0	0	0	95	C:\Users\Jim\AppData\Local\Google\Chrome\User
0.8975765	50	8	8	17	0	454,656	0	4	0	13	C:\Users\Jim\Desktop\putty.exe
0.0004047	32	5	5	17	0	78,477	0	0	0	5	C:\Program Files (x86)\Motorola Media Link\onlinese
0.0807963	15	0	0	15	0	122,880	0	0	0	0	C:\Windows\System32\ExplorerFrame.dll
0.1022874	15	0	0	15	0	282,624	0	0	0	0	C:\Windows\assembly\NativeImages_v2.0.50727_6
0.0003944	28	5	5	13	0	59,056	0	0	0	5	C:\Program Files (x86)\Motorola Media Link\NSC\N
0.0000851	12	0	0	12	0	192	0	0	0	0	C:\Users\Jim\AppData\Roaming\Skype\shared_dy
0.0001011	12	0	0	12	0	12	0	0	0	0	C:\Users\Jim\AppData\Roaming\Skype\shared_dy
0.0352880	10	0	0	10	0	327,680	0	0	0	0	C:\Windows\Microsoft.NET\Framework64\v2.0.507
0.0424288	477	104	104	9	0	106,496	0	0	0	260	C:\Program Files\Microsoft IntelliPoint\ipoint.exe
0.2076150	24	2	3	9	0	114,688	0	0	0	10	C:\Program Files\Microsoft Office\Office14\GROOV
0.0601132	9	0	0	9	0	135,168	0	0	0	0	C:\Windows\winsxs\amd64_microsoft.vc90.mfc_1fc
0.3104952	107	0	0	8	87	36	223,560	0	0	12	C:\Users\Jim\AppData\Roaming\Skype\jimmoyle\m
0.0003274	23	5	5	8	0	36,138	0	0	0	5	C:\Program Files (x86)\Motorola Media Link\smc\SM
0.2104738	8	0	0	8	0	42,496	0	0	0	0	C:\Windows\System32\mswsock.dll

Filter... 915 file paths Save... OK

Windows Boot

The graph shown above was Windows after boot, but before logon. So it would seem a good time to look at the previous step to this, which is Windows boot.



The absolute key thing to look at here is the scale. In the previous graph we had a peak of 65, in this graph the peak is 5,200 IOPS.

This is an unmodified Windows 7 image with 2GB RAM and 2 vCPUs. Allowing the VM access to unlimited IOPS gives us a 12 second boot time.

A 5,200 spike in IOPS seems to be anathema to having allowed 4-50 IOPS per desktop. The advice from Citrix earlier only regards steady state (i.e. post boot and logon). So what do we do about boot time?

Usually booting a VDI environment will happen overnight, a certain number of machines will be pre-booted so that enough instances of windows are available for when staff start in the morning. This process can take many hours to achieve depending on how many machines are booted and how many IOPS are available. You can easily see this effect when you have ~150 IOPS available in a modern PC and it will take 1-2 minutes to boot. Compared to the twelve second boot seen above when > 5000 IOPS are available.

So what happens when a boot is either implemented by the user or the administrator during working hours? This case must be carefully considered as either the option to reboot must be removed from users, or in the case of a stateless desktop which will reboot after the user logs off, the peak must be accounted for by the administrator. On the administrator side, there are situations when a mass boot during working hours is necessary because there might be a need to apply or roll back OS or application patches, there may have been a failure of a Hypervisor or storage array, a security update needs to be pushed to the anti-virus agent or any number of other scenarios. Basically, if you say you will never need to do a mass boot during working hours, you are stating that you know nothing will ever go wrong in the environment.

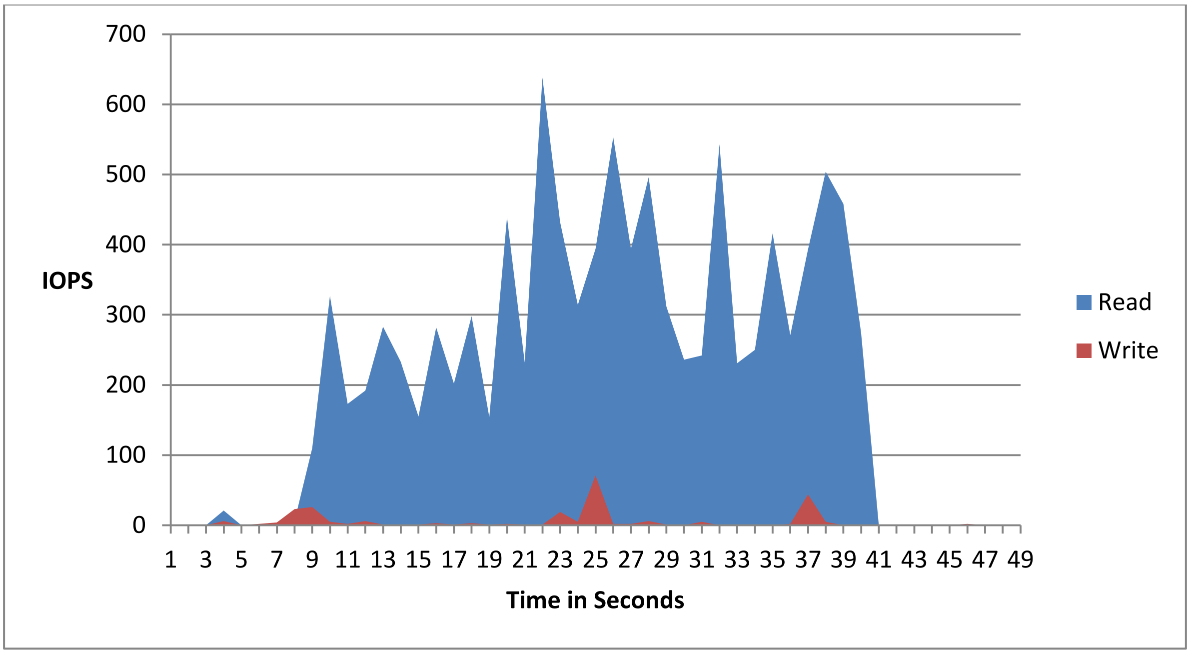
If you do not have the IOPS to manage a quick mass boot within working hours, you need to set the expectation of management and users that their working environment may not be back available to them for several hours after a problem.

I have spoken to clients who don't have the time to pre-boot their machines overnight in batches due to their IOPS limit. This means that the VMs are kept running 24/7, this is obviously not a tenable position e.g. from a datacentre energy perspective you are running resources unnecessarily

Another issue with pre-booting is that if you run a follow the sun solution, where the environment is in use all day and night. This means that you must allow enough resources to cope with mass boots.

In an emergency patch or a disaster recovery situation, we would also need to be able to survive a boot/logon storm. If you don't have enough IOPS to cope, you need to be clear about the SLA you are providing to the business.

The second lower extended peak area of activity shown above and in more detail below is also part of the Windows boot process. Although the ability to logon is available before this process has finished.



Again the scale of the graph must be carefully attended to as this is system applications starting (i.e. VMware tools or Anti-Virus). Other agents and drivers that are installed into the VM will have an increasingly deleterious effect on the performance. This is especially the case where agents 'phone home' for updates again like VMware tools or antivirus agents.

Interestingly, the area under the graph, which shows the total IOPS used is very similar for both the Initial boot and the subsequent agent start.

Initial Boot read IOPS: 10029

Subsequent agent start read IOPs: 10437

So given this information we can work out how long a boot should take if we allow 20 IOPS per VM. I have also included the write IOPS for both stages.

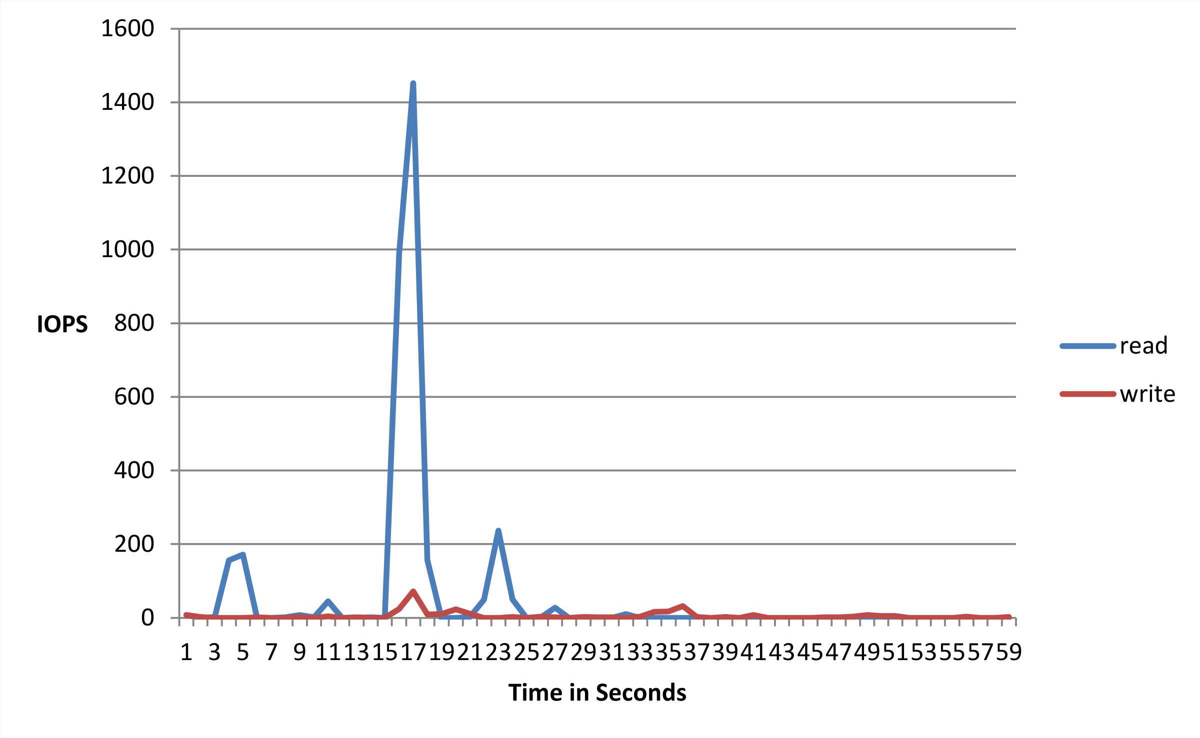
$$\frac{10029\ boot + 10437\ agent + 3668\ write}{20\ IOPS} = 1206\ seconds$$

This will give us greater than a 20 minute boot per VM if we allow 20 IOPS.

Longer than 5 minutes will generally cause some of the Windows instances to fail to boot, due to storage latency, not to mention anger users. In this case, we must allow the VMs to be booted in batches. To try and keep the boot time below five minutes, the VMs in this example would need to be booted in 4- 5 stages. This would take approximately an hour with appropriate gaps between stages to ensure that the previous staged boots completed successfully.

Windows Logon

Windows logon is an area that cannot be ameliorated by pre-booting. Users must log themselves on when they start work, which will generate the logon storm. The chart below is the minimum you can expect from a logon as it consists of the minimum possible start-up applications and a local profile.



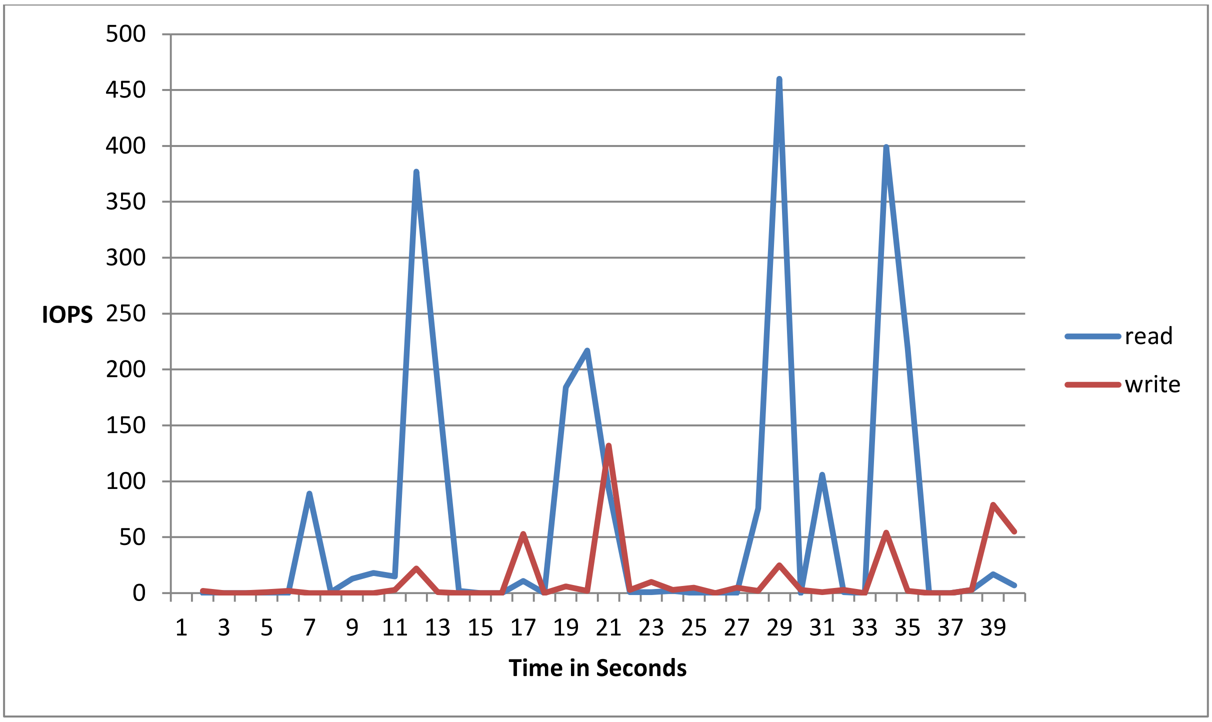
The logon storms that can be experienced depend greatly on what profile solution you have in place, how much software starts upon logon and whether you use application virtualisation. An effective plan for a logon storm is very important, as user experience is a key to acceptance of a VDI project. The very first part of the VDI infrastructure that a user will experience will be the logon and it can lead to an impression that the solution does not perform correctly. The time to login must be the same or better than a physical PC or users will instantly question the performance of the Virtual Desktop Infrastructure.

It is essential that appropriate testing is done with your own image to correctly size for a logon storm.

Help is available to reduce the impact of logon by installing applications in the image, pre-caching virtualised applications, and streaming roaming profiles. Most importantly, either using folder redirection, on-demand profile streaming or a combination of both can be used to help reduce the impact of a logon storm.

Application Launches

Below is the IO load for launching Word, Excel and Outlook. This shows very much the same behaviour as booting does in that it has a very high initial read peak. Application opening times are another key factor in the user identifying the perceived speed of the Solution. If key applications such as a Microsoft Office application take more than a few seconds to launch, users will notice instantly and complain.

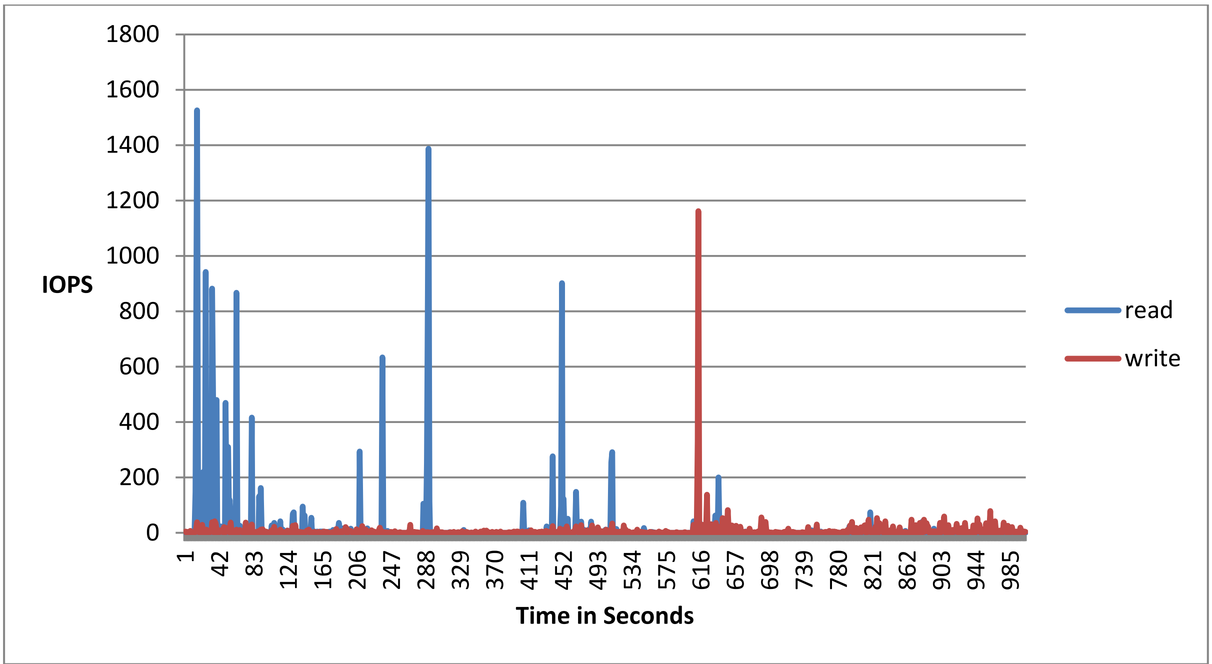

IOPS
500
450
400
350
300
250
200
150
100
50
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39
Time in Seconds
read
write

Steady State

To simulate a steady state workload, the Login VSI 3.0 medium workload was used. As we saw above the initial launch of applications takes a large amount of read IOPS, as applications are re-launched, the IOPS peaks disappear. This is because Windows has cached the applications in memory. You can easily see this effect if you launch an application for the second time on your desktop.

The last third of the graph below is the area which is actually representative of a steady state and is the area that write IOPS is consistently above read IOPS. This is the area that people mean when they say that 80% of Windows IOPS is write and 20% is read in a VDI workload.

In fact, if we ignore the initial peaks and take the graph from 700 seconds onwards, we have 82% write IOPS.

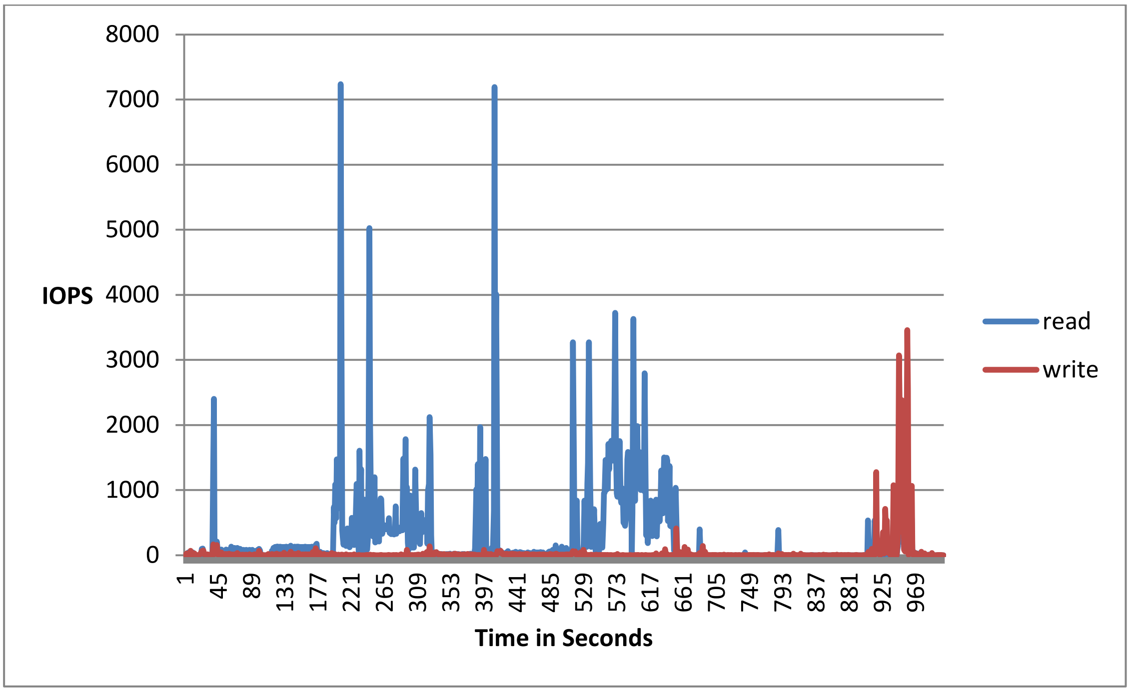


This last third of the graph above is what almost all vendors refer to when sizing for IOPS. As a VDI architect you simply cannot ignore the rest of Windows activity.

Anti-Virus

There has been a lot said about how Anti-Virus has a large impact on VDI. We saw in the boot graph how the start-up of background applications has almost as much impact as the boot itself. One example of the impact of Anti-Virus comes when a scan is launched. Below, we have two examples of Anti-virus full systems scans:

Avira



Total I/O consumed: 309004

Peak Read: 7234

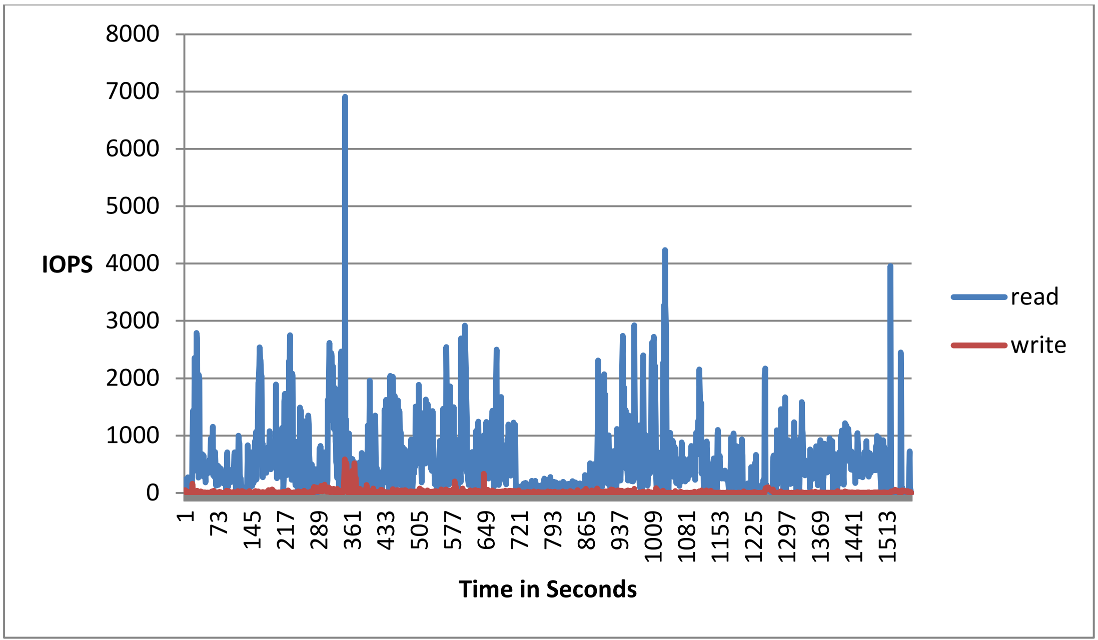
Peak Write: 3459

Time Taken 16 minutes

This shows us that an Anti-Virus scan is much more intensive in terms of I/O than even boot. With the total I/O consumed being 15x more.

To compare, here is a graph of Microsoft Security essentials

Microsoft Security Essentials



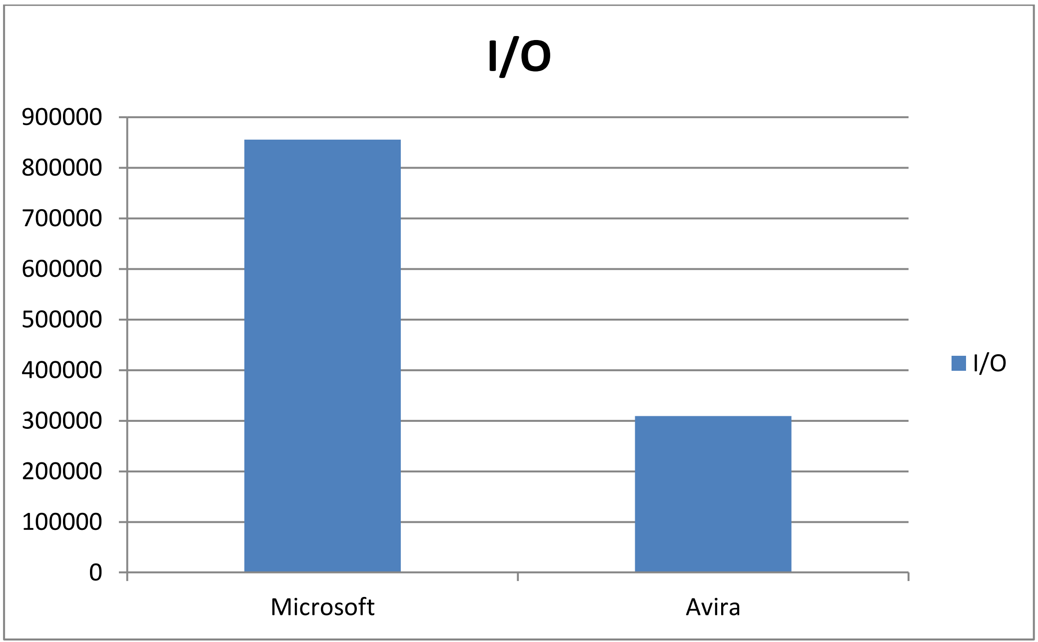
Total I/O consumed: 855553

Peak Read: 6908

Peak Write: 588

Time Taken 26 minutes

We can see here that the Microsoft AV scan consumes over double the I/O of the Avira scan,



There are optimisations you can make to AV to help with these situations . But, this shows that at the very least it is worthwhile comparing the I/O impact of various solutions if you are thinking of deploying them in your VDI environment and that an “average” IOPS value is of little use in designs. AV vendors are starting to develop solutions aimed specifically at targeting VDI environments and we can see this is absolutely necessary. However, none of the solutions will totally remove the I/O hit for AV: at least it will help stop it becoming such a monster. Also, it may take some time to make a full transition from traditional anti-virus as organizations have made it a requirement in their standard image and it is built into compliance along with other endpoint security protections such as personal firewalls.

This again shows us how in a virtual shared environment with a client OS that doesn’t play well with others, one person doing something can impact hundreds of other users.

You could suggest overnight scans to be the answer: but this means you do not have any flexibility in case of disaster. If you need to clean your machines but you cannot start a scan because of the performance hit, it’s not a good solution.

Just for fun, let’s work out if you allowed 20 IOPS per person and you HAD to clean those machines of an infection how long would it take using Microsoft Security essentials

$$\frac{855553\, Total\ IO}{20\, IOPS} = 42777.65\, Seconds$$

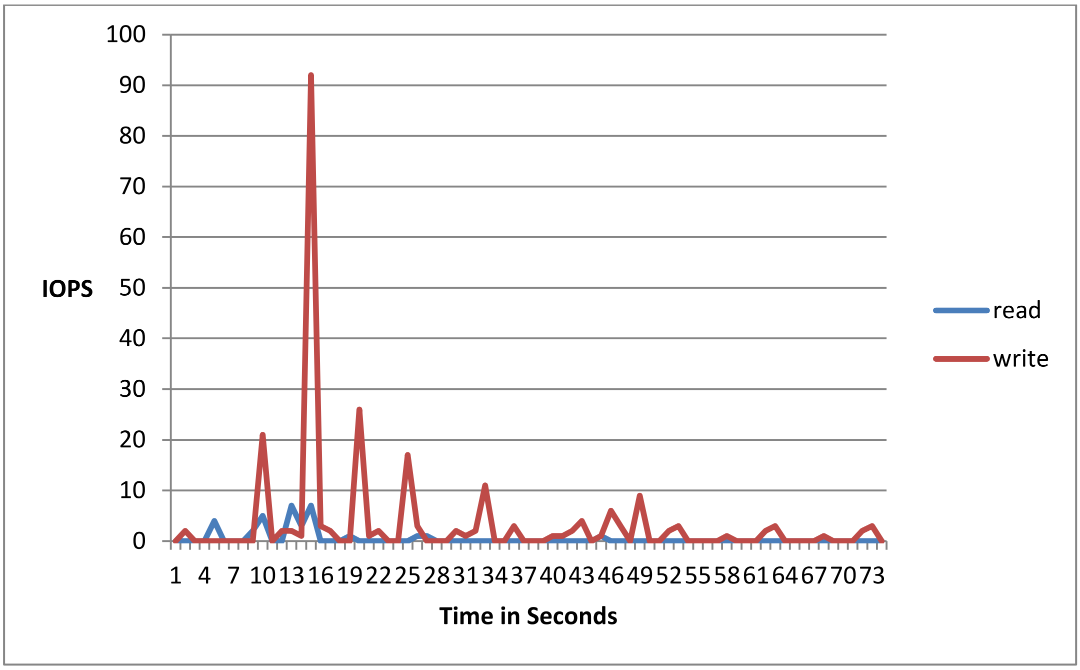
42777.65 Seconds is 11 hours and 53 minutes

Basically you would have to do nothing except do the AV scan and working at full capacity the task would take 12 hours.

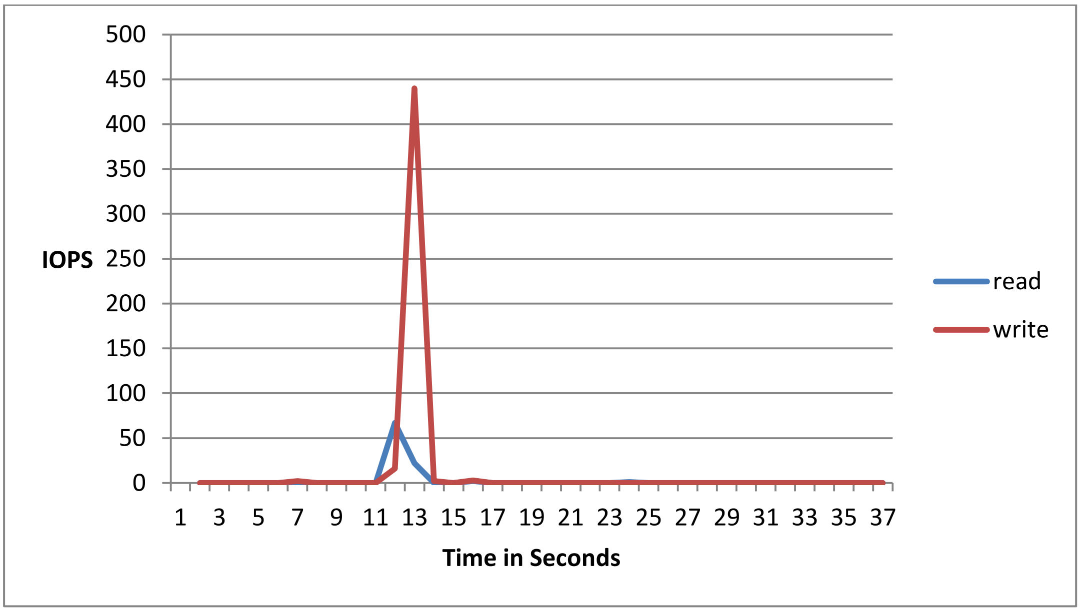
If you want to pre-boot and AV scan when there are no users on the system, now it is looking like a weekend job.

Logoff and Shutdown

Logoff, as might be expected, has a much more significant write peak as the profile is written back down to the disk.



Shutdown again has a large write peak as data is written back down to the disk.



Solutions

Whether your CIO has deemed that they want to deploy once and engineer once for your physical and virtual estate or you have committed to a fully optimised stack with streamed profiles and applications and a stateless VDI image, you will need to plan effectively to deal with IOPS.

Anti-Virus

Anti-virus vendors are starting to develop a VDI software model for VDI. These solutions try to consolidate and offload antivirus to a virtual appliance relieving the pressure on the Windows VM. These solutions are in their infancy and should be used with care. Also, keep in mind that using a different anti-virus solution than you are using in your physical PC image may require security and compliance approvals.

Storage Hardware

You can of course throw spindles at the problem, but as we saw at the start, this is very expensive from both a CAPEX (purchasing the hardware) and OPEX (rack space, power, cooling). I have spoken with organisations that have spent over $1,000USD per desktop to solve the problem this way.

Cache on SAN is a way many storage vendors try to get more IOPS from an existing array of spindles. This works for reads, but not very well for writes. Writes require a very large open cache to work effectively. Also, eventually all writes have to go to the spindles, so if the average writes are greater than the spindles can handle you will be in trouble.

Local SSD drives can work in a stateless model, but of course won't generally work in a persistent model and have limited random write IOPS performance, especially when capacity is nearly full. Also it can be hard to fit enough drives into the server to create the space needed if you have high memory and CPU servers for your VDI solution.

Also on the SSD front, you can buy SSD based storage arrays, these will give you SSD performance for both stateless and persistent desktops, but often suffer from low capacity and can be expensive.

Software

One of the most effective ways to deal with the Windows 7 IOPS problem is to process, deduplicate and optimize the IO generated by the Windows 7 operating system and applications at the hypervisor or rack level before reaching the storage system. This effectively delivers massive amounts of local virtual IOPS to the desktops without adding any storage. Because so much of the IO traffic generated by the desktops is duplicate, up to 90% of the IO can be eliminated from the equation using software.

Atlantis Computing offers just this kind of software optimised for the high I/O load of a Virtual Desktop Infrastructure.

Conclusion

A performant and value for money VDI environment is not hard to achieve with the right information about IOPS and the right technologies to solve the problem.

The examples above are just some of the situations which can trip you up when sizing for IOPS. There are many more. Any application in your enterprise has the potential to create a situation where the IOPS demand goes through the roof, particularly applications with agents. The answer is to monitor the workstations you want to virtualise BEFORE you size your storage. Real information is always better than vendor information. Once you know your IOPS requirement, decide whether you want to do stateless or persistent desktops or a mix of both. This will then give you a steer as to which technologies you need to utilise to help you ensure a successful VDI deployment.

Further Reading and References

Atlantis Computing
http://www.atlantiscomputing.com/win7iops

Citrix Virtual Desktop Resource allocation
http://community.citrix.com/display/ocb/2010/11/12/Virtual+Desktop+Resource+Allocation

IOMeter
http://www.iometer.org/

WinSAT
http://technet.microsoft.com/en-us/library/cc742157(WS.10).aspx

Performance Analysis of Logs (PAL) Tool
http://pal.codeplex.com/

Win7 perfmon
http://technet.microsoft.com/en-gb/library/cc749249.aspx

Logman
http://technet.microsoft.com/en-us/library/cc788121(WS.10).aspx

ESXTop
http://communities.vmware.com/docs/DOC-5490
http://www.yellow-bricks.com/esxtop/

ESX Storage Queues and Performance
http://communities.vmware.com/docs/DOC-6490

XenServer Storage
http://support.citrix.com/article/ctx119088

Understanding Memory Resource Management in VMware ESX 4.1
http://www.vmware.com/files/pdf/techpaper/vsp_41_perf_memory_mgmt.pdf

SSD performance
http://www.ssdperformanceblog.com/2010/07/free-space-and-write-performance/

VDI for developers
http://ultrasub.nl/2011/05/05/vdi-and-iops/