VDI & Storage:
Deep Impact

Author(s) : Herco van Brug

Version: 1.0

Date: December 10, 2009

REFERENCES

Reference	Title
http://www.vmware.com/files/pdf/resources/vmware-view-reference-architecture.pdf	VMware View Reference Architecture
http://h71028.www7.hp.com/ERC/downloads/4AA2-3017ENW.pdf	HP: Implementing a VDI infrastructure
http://support.citrix.com/servlet/KbServlet/download/19754-102-376939/XD%20-%20Design%20Handbook.pdf	Citrix XenDesktop Design Guidelines
http://technet.microsoft.com/en-us/library/cc748650.aspx	Address Space Load Randomization
http://technet.microsoft.com/en-us/library/aa995867(EXCHG.65).aspx http://download.microsoft.com/download/C/E/7/CE7DA506-CEDF-43DB-8179-D73DA13668C5/DiskPartitionAlignment.docx	Windows disk alignment
http://en.wikipedia.org/wiki/Redundant_Array_of_Independent_Disks	Wikipedia: RAID levels
http://en.wikipedia.org/wiki/Scsi	Wikipedia: SCSI
http://en.wikipedia.org/wiki/Integrated_Drive_Electronics	Wikipedia: ATA
http://virtuall.eu/blog/creating-a-vdi-template	Create a VDI image that minimizes IOps

CONTENTS

1. INTRODUCTION

Virtual Desktop Infrastructure, or VDI, is hot. It's cool, secure, centrally managed, flexible - it's an IT manager's dream.

VDI comes in two flavours; Service Hosted VDI (Centralized, single-user remote vDesktop solution) and Client-Side VDI (local, single-user vDesktop solution).

The advantages of a VDI infrastructure are that virtual desktops are hardware independent and can be accessed from any common OS. It is also much easier to deploy virtual desktops and to facilitate the freedom that the users require of them. And because of the single-user OS, application compatibility is much less of an issue than it is with terminal servers.

However, when implementing a VDI infrastructure certain points need to be addressed. First of all, the TCO/ROI calculation may not be as rosy as some people suggest. Secondly, the performance impact on applications, specifically multimedia and 3D applications, needs to be investigated. And finally, don't forget to check licensing aspects, as this can be a very significant factor in VDI infrastructure.

While centralized desktop computing provides important advantages, all resources come together in the datacentre. That means that the CPU resources, memory resources, networking and disk resources all need to be facilitated from a single point - the virtual infrastructure.

The advantage of a central infrastructure is that, when sized properly, it is more flexible in terms of resource consumption than decentralized computing. It is also more capable of handling a certain amount of peak loads, as these only occur once in a while on a small number of systems in an average datacentre.

But what if the peak loads are sustained and the averages are so high that the cost of facilitating them is disproportionate to that of decentralized computing?

As it turns out, there is a hidden danger to VDI. There's a killer named "IOPS".

2. THE CLIENT IO

A Windows client that is running on local hardware has a local disk. This is usually an IDE or SATA disk rotating at 5,400 or 7,200 RPM. At that rate it can deliver about 40 to 50 IOPS.

When a Windows client starts, it loads both the basic OS and a number of services. Many of those services provide functionality that may or may not be needed for a physical system and make life easier for the user. But when the client is a virtual one, a lot of those services are unnecessary or even counter-productive. Indexing services, hardware services (wireless LAN), prefetching and other services all produce many IOPS in trying to optimize loading speed, which works well on physical clients but loses all effectiveness on virtual clients.

The reason for this is that Windows tries to optimize disk IO by making reads and writes contiguous. That means that reading from a disk in a constant stream where the disk's heads move about as little as possible is faster than when the head needs to move all over the disk to read blocks for random reads. In other words, random IOs are much slower than contiguous ones.

The amount of IOPS a client produces is greatly dependant on the services it's running, but even more so on the applications a user is running. Even the way applications are provisioned to the user impacts the IOPS they require.

For light users the amount of IOPS for a running system amounts to about three to four. Medium users show around eight to ten IOPS and heavy users use an average of 14 to 20 IOPS.

Now the most surprising fact; those IOPS are mostly WRITES. A great many researchers have tested the IOPS in labs and in controlled environments using fixed test scripts. The read/write ratio turned out to be as high as 90/10 as a percentage. But in reality users run dozens or even hundreds of different applications, whether virtualized or installed. In practice, the R/W ratio turns out to be 50/50 percent at best! In most cases the ratio is more like 30/70, often even 20/80 and sometimes as bad as 10/90 percent.

But why is that important? Most vendors don't even mention IOPS or differentiate between reads and writes in their reference designs.

3. THE STORAGE IO

When all IOs from a client need to come from a shared storage (attached directly to the virtualization host or through a Storage Area Network) and many clients read and write simultaneously, the IOs are, from the storage point of view, 100 percent random IOs.

3.1 SCSI vs ATA

There are two main forms of disks - SCSI and ATA. Both have a parallel version (regular SCSI vs IDE or PATA) and serial version (SAS vs SATA).

The main differences between the architecture of the SCSI and ATA disks are rotation speed and protocol. To start with the protocol, the SCSI protocol is highly efficient with multiple devices on the same bus, and it also supports command queuing. ATA devices have to wait on each other, making them slower when grouped together.

The higher rotation speed means that when the head needs to move to a different location, it does not need to wait as long for the data to pass beneath it. So a SCSI disk can produce more IOPS than an ATA disk. The faster a disk rotates, the less time the head needs to wait before data passes beneath it and the sooner it can move to the next position, ergo the more IOs it can handle per second.

To give some idea of the numbers involved; a 15,000 RPM disk can handle about 180 random IOPS, a 5,400 RPM disk about 50. These are gross figures and the number of IOPS that are available to the hosts depend very much on the way they are configured together and on the overhead of the storage system. In an average SAN, the net IOPS from 15,000 RPM disks is 30 percent less than the gross IOPS.

3.2 RAID LEVEL

There are several ways to get disks to work together as a group. Some of these are designed for speed, others for redundancy or anything in between.

3.2.1 RAID5

The way a traditional RAID5 system works is that it writes the data across a set of hard disks, calculates the parity for that data and writes that parity to one of the hard disks in the set. This parity block is written to a different disk in the set for every further block of data.

To write to a RAID5 system, the affected blocks are first read, the changed data is inputted, the new parity is calculated and the blocks are then written back. On systems with large RAID5 sets this means a write IO is many times slower than a read IO. Some storage systems, like HP's EVA, have a fixed set of four blocks for which parity is calculated, no matter how many disks are in a group. This increases overhead on a RAID5 group because every set of four disks needs a fifth one, but it does speed things up. Also, on most storage systems, write operations are written to cache. This means that writes are acknowledged back to the writing system with very low latency. The actual write to disk process takes place in the background. This makes

incidental write operations very speedy, but large write streams will still need to go directly to disk.

With 15,000 RPM disks the amount of read IOPS are somewhere in the 150-160 range while write IOPS are closer to the 35-45 range.

3.2.2 RAID1

A RAID1 set is also called a mirror. Every block of data is written to two disks and read from either one. For a write IO to occur, the data doesn't need to be read first because it does not change part of a parity set of blocks but rather just writes that single block of data. This means that writing to a RAID1 is much faster than to a RAID5.

With RAID1 the data is read from one of the two disks in a set and written to both. So for 15,000 RPM disks, the figures for a RAID1 set are still 150-160 IOPS for reads, but 70-80 for writes.

3.2.3 RAID0

RAID0 is also called striping. Blocks of data are written in sequence to all disks in a RAID0 set but only to one at the time. So if one disk in the set fails, all data from the set of disks is lost. But because there is no overhead in a RAID0 set, it is the fastest way of reading and writing data. In practice this can only be used for volatile data like temporary files and temporary caches, and also perhaps for pagefiles.

If used, the amount of IOPS a RAID0 set can provide with 15,000 RPM disks is 150-160 for reads and 140-150 for writes.

3.2.4 RAID-DP

RAID-DP is a special version of RAID4 in the sense that it uses two instead of one parity disks. RAID4 is like RAID5 except that, instead of spreading parity across all disks, the parity is only written to one disk. RAID-DP uses two parity disks that contain the same data, so that failure of one disks does not require a rebuild of the parity (very storage- and CPU-intensive). This way, RAID-DP has the ability to survive the loss of any two disks. When a parity disk fails, a new disk simply needs to replicate the data from the other parity disk.

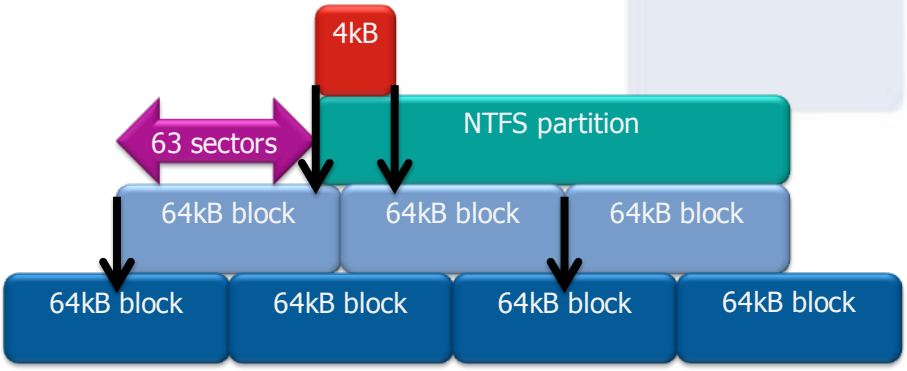
This technology is used with great efficiency in NetApp storage. The way the NetApp underlying filesystem works means that the data for RAID-DP doesn't need to be read first before it can be written, making it as fast as RAID10 but with a level of resilience similar to RAID6.

So, with 15,000 RPM disks in a RAID-DP, the number of read IOPS per disk is some 150-160 but the number of write IOPS lies somewhere between 70-80 IOPS.

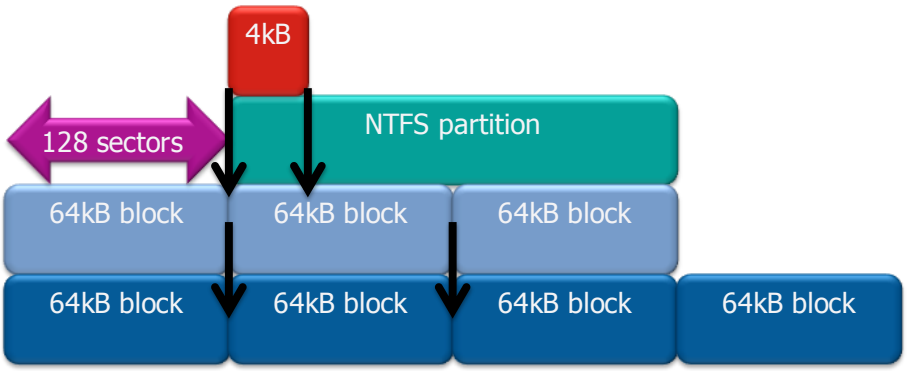
3.3 DISK ALIGNMENT

Because we want to minimize the amount of IOPS from the storage we want every IO to be as efficient as possible. Disk alignment is an important factor in this.

Not every byte is read separately from the storage. From a storage perspective, the data is split into blocks of 32 kB, 64 kB or 128 kB, depending on the vendors. If the filesystem on top of those blocks is not perfectly aligned with the blocks, an IO from the filesystem will result in 2 IOs from the storage system. If that filesystem is on a virtual disk and that virtual disk sits on a filesystem that is misaligned, the single IO from the client can result in three IOs from the storage. This means it is of utmost importance that all levels of filesystems are aligned to the storage.



Unfortunately, Windows XP and 2003 setup process misalign their partition by default by creating a signature on the first part of the disk and starting the actual partition at the last few sectors of the first block, misaligning the partition completely. To set this up correctly, create a partition manually using 'diskpart' or a Linux 'fdisk' and put the start of the partition at sector 128. A sector is 512 bytes, putting the first sector of the partition precisely at the 64 kB marker. Once the partition is aligned, every IO from the partition results in a single IO from the storage.



The same goes for a VMFS. When created through the ESX Service Console it will, by default, be misaligned. Use fdisk and expert mode to align the VMFS partition or create the partition through VMware vCenter which will perform the alignment automatically.

Windows Vista and later versions try to properly align the disk. By default they align their partition at 1 MB, but it's always a good idea to check if this actually is the case[1].

The gain from aligning disks can be 3-5 percent for large files or streams up to 30-50 percent for small (random) IOs. And because a VDI IO is an almost completely random IO, the performance gain from aligning the disks properly can be substantial.

3.4 Prefetching and Defragging

The NTFS filesystem on a Windows client uses 4 kB blocks by default. Luckily, Windows tries to optimize disk requests to some extent by grouping block requests together if, from a file perspective, they are contiguous. That means it is important that files are defragged. However, when a client is running applications, it turns out that files are for the most part written. If defragging is enabled during production hours the gain is practically zero, while the process itself adds to the IOs. Therefore it is best practice to disable defragging completely once the master image is complete.

The same goes for prefetching. Prefetching is a process that puts all files read more frequently in a special cache directory in Windows, so that the reading of these files becomes one

[1] A quick way to check if a partition is aligned is by typing "wmic partition get BlockSize, StartingOffset, Name, Index" in a command shell. If the number isn't a multiple of 65536 (64 kB) or 1048575 (1 MB) the partition is unaligned.

contiguous reading stream, minimizing IO and maximizing throughput. But because IOs from a large number of clients makes it totally random from a storage point of view, prefetching files no longer matters and the prefetching process only adds to the IOs once again. So prefetching should also be completely disabled.

If the storage is de-duplicating the disks, moving files around inside those disks will greatly disturb the effectiveness of de-duplication. That is yet another reason to disable features like prefetching and defragging.

4. THE MATHS

So much for the theory. How do we use this knowledge to properly size the infrastructure?

4.1 PROCESSOR

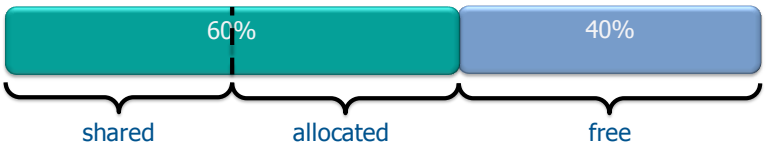
On average, a VDI client can share a processor core with six to nine others. Of course, everything depends on what applications are being used, but let's take an average of 7 VMs per core. With a dual socket, quad core CPU system that means we can house 7 x 2 x 4 = 56 clients. However, the Intel Nehalem architecture is very efficient with hyper-threading and allows 50-80 percent more clients. That means that when it comes to the CPU, we can host 150% * 56 = 84 VMs.

4.2 MEMORY

The amount of memory the host must have depends primarily on the applications the users require and the OS they use. On average a Windows XP client needs 400-500 MB of RAM for basic operations and a standard set of applications. Add some caching and the memory usage should stay below 700 MB.

The Windows OS starts paging when 75 percent of its memory is allocated. It will always try to keep at least 25 percent free. But paging in virtual environments is a performance-killer. So instead of giving it the recommended (in physical systems) amount of 1.5 to 2 times the amount of memory in swap space, we limit the pagefile size to a fixed amount of 200 to perhaps 500 MB. If that is not enough, just add more RAM to the client, rather than extending the pagefile.

This also means we aim for at least 25 percent free RAM space with most applications running. Additionally, about half of the used memory contains the same blocks in all clients (Windows DLLs, same applications, etc). This is lower on Windows 7 clients because of ASLR (Address Space Load Randomization), which means that the amount of memory shared between clients is 25% (empty space) + 75% / 2 = 62.5%.



So when running Windows XP on ESX servers, if 60 percent of memory per client is actually being used, 50 percent of which is shared between clients, we need 1 GB x 60% x 50% = 300 MB per client. Every VM needs about 5 percent more than the amount allocated as overhead from the host. So you need an additional 50 MB (5 percent of 1 GB) per client.

We have seen from the CPU calculation that we can host 84 clients, so a host would need 4 GB (for the host itself) + 350 MB x 84 = at least 34 GB of RAM.

However, if 75 percent of memory is used and only a third of that can be shared, every client needs 1 GB x 75% x 67% = 512 MB of dedicated host memory. So for 84 clients the host needs 4 GB + (512 + 50) MB x 84 = 52 GB of RAM.

Of course if you run on a host that doesn't support transparent page sharing, the amount of memory needed is 4 GB + 84 * (1024 + 50) MB = 96 GB of RAM.

For Windows 7 clients the numbers are (2 GB + 100 MB) x 60% x 50% = 660 MB per client on average, 4 GB + 660 MB x 84 = 60 GB of minimum host memory and 4 GB + 84 x (2 GB + 100 MB) = 188 GB per host if the host doesn't support memory over-commitment.

4.3 DISKS

The amount of IOPS a client produces is very much dependant on the users and their applications. But on average, the IOPS required amount to eight to ten per client in a read/write ratio of between 40/60 percent and 20/80 percent. For XP the average is closer to eight, for Windows 7 it is closer to ten, assuming the base image is optimized to do as little as possible by itself and all IOs come from the applications, not the OS.

When placing 84 clients on a host, the amount of IOPS required would be 840, of which 670 are writes and 170 are reads. To save on disk space, the disks are normally put in a RAID5 set up. But to deliver those numbers, we need 670 / 45 + 170 / 90 (see 'RAID5' section earlier in this document) = 17 disks per host. Whether or not this is put in a central storage system or as locally attached storage, we will still require 17 disks for 84 VMs. If we used RAID1, the number changes to 670 / 90 + 170 / 110 = 9 disks. That means, however, that using 144 GB disks, the net amount of storage drops from 17 x 144 GB x 0.8 (RAID5 overhead) = 1960 GB to 9 x 144 GB x 0.5 (RAID1 overhead) = 650 GB.

4.4 PRACTICAL NUMBERS

All these numbers assume that clients are well-behaved and that most of the peaks are absorbed in the large averages. But in reality you may want to add some margins to that. To be on the safe side, a more commonly used number of clients per host is 65 (about 3/4 of 84). That means that the *minimum* amount of memory for the average XP client solution would be 65 x 350 MB + 4 GB = 27 GB, or for Windows 7: 65 x 660 MB + 4 GB = 47 GB.

The amount of IOPS needed in these cases is 10 IOPS x 65 clients = 650 IOPS where 80 percent (= 520) are writes and 20 percent (= 130) are reads. With RAID5 that means we need (520 / 45) + (130 / 80) = 13 disks for every 65 clients. Should you require 1,000 VDI desktops, you will need (1000 / 65) x 13 = 200 disks. Put on RAID1, that number decreases to 108, which is quite substantial considering that it is still only nine clients per disk.

So, to be sure of the number you need to use, insist on running a pilot with the new environment where a set of users actually use the new environment in production. You can only accurately size your infrastructure once you see the numbers for those users, the applications they use and the IOPS they produce. Too much is dependent on correct sizing - especially in the storage part of the equation!

5. SUMMARY

The table below summarizes the sizing parameters:

Setting	Sizing
Number of VDI clients per CPU core	6-9
Hyper-threading effectiveness on Intel Nehalem systems	150 – 180%

Setting	Sizing
Amount of memory per VDI client	Windows XP: 1 GB Windows 7: 2 GB
Amount of memory actually allocated (in actual host memory)	Minimum: 37.5% Average: 50% Not shared: 100%
Number of clients per host	65
Minimum amount of memory per host	XP: 27 GB, W7: 47 GB
Average amount of memory per host	XP: 37 GB, W7: 76 GB
Amount of memory per host (if not shared)	XP: 76 GB, W7: 144 GB
IOPS per VDI client	Light user: 3-4-5 Medium user: 6-8-10 Heavy user: 14-20

The following table summarizes the IOPS for the different RAID solutions:

Raid level	Read IOPS 15k	Write IOPS 15k	Read IOPS 10k	Write IOPSs 10k
RAID 5	150-160	35-45	110-120	25-35
RAID 1	150-160	70-80	110-120	50-60
RAID 0	150-160	140-150	110-120	100-110
RAID DP	150-160	70-80	110-120	50-60

To illustrate the above figures, a few samples follow:

Scenario with 65 clients/host	10 IOPS R/W 20/80%	10 IOPS R/W 50/50%	5 IOPS R/W 20/80%	5 IOPS R/W 50/50%
VDI clients per disk	RAID5: 5 RAID1: 9 RAID0: 15 RAIDDP: 9	RAID5: 7 RAID1: 10 RAID0: 16 RAIDDP: 10	RAID5: 10 RAID1: 17 RAID0: 30 RAIDDP: 17	RAID5: 14 RAID1: 21 RAID0: 31 RAIDDP: 21
Number of disks per host	RAID5: 13 RAID1: 8 RAID0: 5 RAIDDP: 8	RAID5: 10 RAID1: 6 RAID0: 4 RAIDDP: 6	RAID5: 7 RAID1: 4 RAID0: 3 RAIDDP: 4	RAID5: 5 RAID1: 3 RAID0: 2 RAIDDP: 3

6. ALTERNATIVES

6.1 CACHE

There are many solutions out there that claim to speed up the storage by multiple factors. NetApp has its Performance Acceleration Module (PAM), Atlantis Computing has vScaler, and that's just the tip of the iceberg. Vendors such as Citrix with its Provisioning Server and VMware with its View Composer also aid storage by single-instancing the main OS disk, making it much easier to cache it.

But in essence they are all read caches. Caching the IOPS for the 30 percent that are reads, even with an effectiveness of 60 percent, will still only cache 30% x 60% = 18% of all IOs. All write IOs still need to go to disk.

However, most storage systems also have 4 GB, 8 GB or more cache built-in. While the way it is utilised is completely different for each vendor and solution, most have a fixed percentage of the cache reserved for writes, and this write cache is generally much smaller than the read cache.

The fact is that when the number of writes remains below a certain level, most of them are handled by cache. Therefore it is fast; much faster than for reads. This cache is, however, only a temporary solution for handling the occasional write IO. If write IOs are sustained and great in number, this cache needs to constantly flush to disk, making it practically ineffective. Since,

with VDI, the large part of the IOs are write IOs, we cannot assume the cache will fix the write IO problems, and we will always need the proper number of disks to handle the write IOs.

6.2 SSD

SSD disks are actually more like large memory sticks rather than disks. The advantage is that they can handle an amazing amount of IOPS; sometimes as high as 50,000 or 100,000. They have no moving parts so accessing any block of data takes mere microseconds, instead of milliseconds.

However, the current state of the SSD drives only allows every cell to be written 1,000 to 10,000 times. That means that, even with smart tricks like moving cells around to spread writes, the sustained writes of a VDI solution would break an SSD disk within a few months. This 'spreading writes around' is called TRIM and is the reason why writes are so much slower than reads on SSDs.

Also, the current backend of any storage solution handle the number of IOPS those drives can offer. Most vendors don't recommend SSD drives as yet for large scale storage demands. Aside from this fact, they are also very expensive - sometimes costing four to ten times as much as 15,000 RPM SCSI disks.

It is expected that this may change soon, as better SSD cells are constantly being developed. With a more even read/write ratio, a longer lifespan, larger disks and better pricing, we may see SSD disks in a SAN become more common within a year or two.

7. IN CONCLUSION

It should be obvious by now that calculating the amount of storage needed in order to properly host VDI is not to be taken lightly. The main bottleneck at the moment is the IOPS. The read/write ratio of the IOPS that we see in practice in most of the reference cases demonstrate figures of 40/60 percent, sometimes even as skewed as 10/90 percent. The fact is that they all demonstrate more writes than reads. And because writes are more costly than reads - on any storage system - the number of disks required increases accordingly, depending on the exact usage of the users and the application.

Some questions remain:

- What is the impact of application virtualization on the R/W IOPS?
- What exactly is the underlying cause of the huge difference in read/write ratios between lab tests and actual production environments?
- What if all the write IOs only need to be written to a small part of the total dataset (such as temporary files and profile data)? Could all the data, or at least most of it, be captured in a large write cache?

These questions will be investigated as an increasing number of VDI projects are launched.

And as a final note, it is imperative that you run a pilot. Run the actual applications with actual users in the production environment beforehand so that you know how they behave and what the read/write ratio is. If you don't size correctly, everybody will complain. All users, from IT staff to management and everybody in between, will complain and the VDI project… will FAIL.

8. ABOUT

8.1 ABOUT PQR

PQR is the specialist for professional ICT infrastructures with a focus on server and storage, virtualization and application availability.

PQR stands for simplicity, freedom and professionalism. We provide our clients with innovative ICT solutions that ensure that application availability and management are optimizedal. We have traceable references and a wide range of expertise in the field, as witnessed by our many high-status partnerships and certifications.

PQR is an HP GOLD Preferred Partner 2009, HP Enterprise Specialist Partner 2007/2008, VMware Premier Partner and Gold Authorized Consultant Partner, Citrix Platinum Solution Advisor, Microsoft Gold Partner Advanced Infrastructures & Security, RES Platinum Partner, NetApp Platinum Partner, Cisco Partner, CommVault Value Added Reseller, HP ProCurve Master Partner, Platespin Platinum Partner and Websense Platinum Partner.

8.2 ABOUT THE AUTHOR

Herco van Brug was born in 1968 and studied mechanical engineering at the University of Twente in the Netherlands. Immediately after graduation he started working at Rijnhaave, later Syntegra. When Syntegra was taken over by British Telecom his position shifted to that of technical specialist, focussing mainly on specialized solutions and migrations.

At present he is a Solutions Architect at PQR, with his primary focus being business continuity solutions in the datacentre. He is the co-author of the Data & System Availability diagram and is certified for Microsoft, RedHat, Citrix and VMware, while as a VMware Authorized Consultant, he undertakes VMware branded Professional Services assignments. He has been a speaker at several national conferences and published a number of articles, all related to virtualization.

PQR B.V.
Rijnzathe 7
3454 PV De Meern
The Netherlands

Tel: +31 (0)30 6629729
Fax: +31 (0)30 6665905
E-mail: info@pqr.nl
www.PQR.com