

Masterproef

Hardware Assisted Virtualization

Studiegebied

Industriële wetenschappen en technologie

Opleiding

Master in de industriële
wetenschappen: Elektronica-ICT

Afstudeerrichting

Multimedia en Informatietechnologie

Academiejaar

2007-2008

Tijl Deneut

Masterproef

Hardware Assisted Virtualization

Studiegebied

Industriële wetenschappen en technologie

Opleiding

Master in de industriële
wetenschappen: Elektronica-ICT

Afstudeerrichting

Multimedia en Informatietechnologie

Academiejaar

2007-2008

Tijl Deneut

VOORWOORD

Deze masterproef was voor mij meer een logische verder zetting van mijn eerdere stage dan een bron van inhoud voor mijn thesis. Mijn achtergrond betreft namelijk een graduaats-opleiding MCT binnen het PIH. Die Multimedia- & Communicatie-technologie opleiding had als orgelpunt een 12 weken durende stage. Deze stage ging ook volledig door binnen het Sizing Servers lab.

Het Sizing Servers lab (www.sizing servers.be) is zeer bekend, niet alleen binnen het PIH maar ook op internationaal vlak. Een prachtige springplank voor een verdere carrière. En dat is uiteindelijk waarvoor elk van ons naar school gaat.

Ik had graag eerst en vooral mijn promotor Johan De Gelas bedankt. Ik besef dat dit voor hem niet het leukste jaar was doordat hij (meer dan hem lief was) voor het papierwerk moest zorgen, zodat we allen tijdig het testmateriaal tot onze beschikking hadden. Dit lukte hem overigens wonderwel: nooit eerder zag het lab zo'n invoer van volledige server-systemen en rand materiaal. Dit alles maakt dat deel uitmaken van dit project een waar geschenk was. Ook de interne promotor Johan Beke verdient een dankwoord. Hij toonde zeker interesse en zat mij net genoeg op de hielen om ervoor te zorgen dat de deadlines werden gehaald, anderzijds bleef hij genoeg aan de kant om de eigenheid van de masterproef te garanderen.

Er werd mij ook de kans gegeven om mijn deleger-capaciteiten uit te bouwen doordat ik 3 mensen mocht wegwijs maken binnen het Sizing Server lab: Aswin Coolsaet, Thomas De Ly en Philip Dubois zijn drie zeer sterke persoonlijkheden die in hun laatste MCT-jaar ook voor het Sizing Server lab gekozen hebben als stage. Hun input voor mijn thesis was van onschatbare waarde en tevens waardeer ik ook de talloze correcties, zowel op mondeling als op schriftelijk gebied.

De andere medewerkers van het lab: Liz Van Dijk, Dieter Vandroemme, Tjerk Ameel & Yves Wouters maakten van deze stage telkens een plezier om te werken (of soms: ontspannen). Tot slot wens ik de morele steun zeker niet te verwaarlozen, goede en kwade dagen heeft iedereen wel eens en op het einde van de rit moet deze thesis best ook nog eens worden overlezen. Voor deze en nog veel meer zaken kon ik stevast rekenen op mijn vriendin Tine Delaere. Ook mijn ouders speelden een belangrijke rol in het succesvol afsluiten van deze gedenkwaardige masterproef.

INHOUDSOPGAVE

VOORWOORD	I
INHOUDSOPGAVE	II
AFKORTINGEN EN VERKLARENDE WOORDENLIJST	V
FIGUURLIJST	VI
TABELLIJST	X
1. Inleiding	1
2. Shared Storage	2
2.1. Inleiding	2
2.1.1. Storage Hardware, kennismaking met SAS	2
2.1.2. DAS, NAS & SAN	3
2.2. SAN: Fibre Channel of iSCSI	4
2.2.1. Fibre Channel	4
2.2.2. iSCSI	6
2.3. Prestatiefactoren	8
2.4. Concrete opbouw	10
2.4.1. Microsoft iSCSI Software Target (WinTarget)	11
2.4.2. Aanmelden met de Microsoft iSCSI Initiator	18
2.4.3. Benchmark opzetten (SQLIO)	23
2.5. Testen en resultaten	26
2.5.1. Opzet, testfactoren	26
2.5.2. Testresultaten	28
3. Hardware Assisted Virtualization, de toekomst vandaag	34
3.1. De theorie: wat is het en hoe werkt het?	34
3.2. Binary Translation (Software Virtualisatie)	36
3.3. Para-virtualisatie	37
3.4. Hardware virtualisatie: Intel VT-x & AMD SVM	38
3.4.1. Eerste generatie	38
3.4.2. Tweede generatie: Nested of Extended Page Tables	41
4. VMware ESX, marktleider in virtualisatie	45

4.1.	VMware Infrastructure 3i	45
4.2.	Concrete installatie van een ESX Host.....	50
4.3.	Configuratie, optimalisatie van een ESX-server	55
4.3.1.	Virtuele schijven op aparte schijf.....	56
4.3.2.	Processor affiniteit.....	58
4.3.3.	VMware Tools.....	59
4.3.4.	Partitie alignering	60
5.	Xen, kostenloze virtualisatie.....	61
5.1.	Werking	61
5.2.	Xen installatie	62
5.2.1.	Algemeen	62
5.2.2.	Driver domains	65
5.2.3.	Een concrete installatie (onder SUSE Linux Enterprise Server 10 SP1)	65
5.3.	Xen eigenschappen en optimalisaties	66
5.3.1.	Xen (domU) systeem dupliceren.....	66
5.3.2.	Voor- en nadelen	67
5.3.3.	Virtuele schijven op aparte schijf.....	67
5.3.4.	Logical Volume Manager.....	68
6.	Microsoft Hyper-V, last but not least.....	69
6.1.	Installatie	69
6.1.1.	Vereisten.....	69
6.1.2.	Concreet	70
6.2.	Eigenschappen en optimalisatie.....	76
6.2.1.	Fysieke schijf gebruiken.....	77
6.2.2.	Integration Services.....	78
7.	Benchmarking Virtualisatie: theorie in de praktijk met resultaten	81
7.1.	Opzet, testfactoren	82
7.1.1.	Prestatiefactoren voor Sysbench	84
7.1.2.	Prestatiefactoren voor MySQL.....	84
7.1.3.	Prestatiefactoren voor SUSE Linux Enterprise Server 10 SP1 x64.....	85
7.1.4.	Prestatiefactoren voor het RAID-systeem.....	85

7.2.	Concreet: opzetten van een test met Sysbench	86
7.2.1.	SLES10 installatie	86
7.2.2.	MySQL installatie	87
7.2.3.	Sysbench installatie	88
7.2.4.	RAID-preparatie	88
7.2.5.	Een test starten	88
7.3.	Testresultaten	90
7.3.1.	Bewijzen, algemene tests	90
7.3.2.	Barcelona tests	91
7.3.3.	Clovertown tests	93
7.3.4.	Harpertown tests	95
7.3.5.	Platform vergelijking	96
7.3.6.	Real World testscenario's	97
8.	Conclusie & besluiten	100
9.	Case Studies	102
9.1.	Case Study 1: MCS VMware uitbreiding	102
9.2.	Case Study 2: Savaco High Availability solutions	109
	LITERATUURLIJST	XI
	Appendix A: SATA/SAS bekabelings Overzicht	A
	Appendix B: Eerste ervaringen met Open-e	C
	Appendix C: Enkele voorbeelden van Storage Appliances	I
	Appendix D: Hardware Overzicht	L
	Appendix E: ESX Troubleshooting	O
	Appendix F: Windows Server 2008 Overzicht	R
	Appendix G: Projectfiche	W

AFKORTINGEN EN VERKLARENDE WOORDENLIJST

GPT	GUID Partition Table, tegenhanger van MBR, een onderdeel van een (logische) schijf waar o.a. de partitiegegevens opgeslagen worden.
MBR	Master Boot Record, tegenhanger van GPT, tevens een deel van een (logische) schijf waar partitiegegevens worden opgeslagen, beperkt tot partities van 2TB.
FBDIMM	Fully Buffered Dual In-line Memory Module, dit is geheugen dat nog een kleine extra chip heeft waar veel gebruikte gegevens worden gebufferd.
ECC	Error Checking & Correcting, een type geheugenmodule dat sector fouten in geheugen kan opvangen en herstellen.
15K5 RPM	15.500 rounds-per-minute of toeren-per-minuut, aanduiding voor de snelheid. Veel gebruikt bij schijven.
SATA	Serial Advanced Technology Attachment, tegenhanger van PATA (Parellel). Is een protocol om data-gegevens serieel te versturen. Gebruikt voor zowel schijven als optische lezers.
x86	Verzamelnaam voor alle recente CISC processors voornamelijk gebruikt voor PC's.
Affiniteit	Het toewijzen van een CPU-kern aan een bepaald proces.
Consolidatie	Samensmelten van meerdere kleine systemen naar één krachtige server.

FIGUURLIJST

<i>Figuur 1: Fysiek verschil tussen SAS & SATA schijfinterface</i>	2
<i>Figuur 2: Het is mogelijk 2 systemen een SAS loopwerk te laten gebruiken, in theorie is zelfs dubbele doorvoersnelheid mogelijk</i>	2
<i>Figuur 3: Schema met overzicht van alle verschillende opslagtechnologieën</i>	3
<i>Figuur 4: Fibre Channel Switch (LightSand S-8100A) ≈ \$ 5,0000.00</i>	4
<i>Figuur 5: Glasvezel LC-LC Kabel Duplex, 1 zendkanaal en 1 ontvangkanaal</i>	5
<i>Figuur 6: Fibre Channel Host Bus Adapter PCI-Express 4X</i>	5
<i>Figuur 7: Promise vTrak E310f Fibre Channel Storage Array</i>	5
<i>Figuur 8: Intel SSR212MC2</i>	7
<i>Figuur 9: Zelfgemaakt schema ter verduidelijking van het partition offset probleem</i>	10
<i>Figuur 10: iSCSI Target Software: Er is een 64bit & een 32bit versie</i>	12
<i>Figuur 11: Opzetten iSCSI Target</i>	13
<i>Figuur 12: Naamgeving iSCSI Target</i>	13
<i>Figuur 13: iSCSI initiator toevoegen aan target</i>	14
<i>Figuur 14: iSCSI Initiator, DNS Naam kiezen</i>	15
<i>Figuur 15: iSCSI Virtual Disk maken</i>	16
<i>Figuur 16: iSCSI Virtual Disk wizard, naamgeving</i>	16
<i>Figuur 17: iSCSI Virtual Disk wizard, grootte</i>	17
<i>Figuur 18: Hier blijven enkel nog een druk op de “OK” & de “Finish” knop over</i>	18
<i>Figuur 19: iSCSI Initiator, IQN naam</i>	19
<i>Figuur 20: iSCSI Portal Discovery</i>	20
<i>Figuur 21: iSCSI Initiator, MPIO aanmaken</i>	21
<i>Figuur 22: iSCSI Initiator, Multi-Path Support</i>	22
<i>Figuur 23: iSCSI Initiator, MPIO opties</i>	23
<i>Figuur 24: SQLIO, schijfbeheer</i>	24
<i>Figuur 25: SQLIO, Diskpart Tool</i>	24
<i>Figuur 26: iSCSI Initiator NTFS, Clustergrootte instellen</i>	25
<i>Figuur 27: Dit is het hoofdvenster van IOMeter</i>	25
<i>Figuur 28: Testresultaten; Partitie offset</i>	28
<i>Figuur 29: Testresultaten: FC tegenover iSCSI; Random Write</i>	29
<i>Figuur 30: Testresultaten: FC tegenover iSCSI; Random Read</i>	29
<i>Figuur 31: Testresultaten: SLES Cache, Random Write</i>	30
<i>Figuur 32: Testresultaten: SLES Cache, Random Read</i>	30
<i>Figuur 33: Testresultaten: Starwind Access modes</i>	31
<i>Figuur 34: Testresultaten: Jumbo-frames</i>	32
<i>Figuur 35: Testresultaten: Netwerk-tuning</i>	33
<i>Figuur 36: Ring privileges met software virtualisatie, de gast besturingssystemen (Guest OS) draaien niet langer in kernel mode (Ring 0), maar met minder rechten in Ring 1</i>	35
<i>Figuur 37: User code (ring 3) wordt rechtstreeks uitgevoerd. Binary Translation gebeurt enkel bij kernel code</i>	36

<i>Figuur 38: Simpele drivers werken met de “normale” Linux drivers in VM0</i>	38
<i>Figuur 39: Hardware Virtualization; het gast systeem staat terug waar het hoort: Ring 0....</i>	39
<i>Figuur 40: VMentry & VMexit latency in de Xeon familie is verlaagd doorheen de jaren.</i>	40
<i>Figuur 41: VMentry & VMexit nummers (in nanoseconden) voor verschillende intel families.</i>	41
<i>Figuur 42: Schematische voorstelling van MMU, TLB, CPU & geheugen.</i>	42
<i>Figuur 43: Virtualisatie van Virtueel Geheugen, Shadow Page Tables</i>	42
<i>Figuur 44: Nested/Extended Page Tables</i>	43
<i>Figuur 45: Overzicht van VMware server producten: VMware Infrastructure.....</i>	45
<i>Figuur 46: VMware ESX.....</i>	46
<i>Figuur 47: Klassieke opstelling VMware Infrastructure</i>	47
<i>Figuur 48: Virtualcenter werking</i>	48
<i>Figuur 49: Virtualcenter screenshot met cluster “SSlab”</i>	48
<i>Figuur 50: VMware Virtualcenter kaart van verschillende hosts.....</i>	50
<i>Figuur 51: ESX-setup: start-scherm</i>	51
<i>Figuur 52: ESX-setup: Partitionering.....</i>	52
<i>Figuur 53: ESX-setup: Bootloader</i>	53
<i>Figuur 54: ESX-setup: Netwerkconfiguratie.....</i>	54
<i>Figuur 55: ESX-setup: Gestart scherm</i>	55
<i>Figuur 56: VMware Infrastructure Client</i>	55
<i>Figuur 57: ESX-optimalisatie: Storage Adapters</i>	56
<i>Figuur 58: ESX-optimalisatie: Networking</i>	57
<i>Figuur 59: ESX-optimalisatie: iSCSI Settings</i>	58
<i>Figuur 60: ESX-optimalisatie: Processor affiniteit</i>	59
<i>Figuur 61: XEN HVM: 32bit- & 64bit-ondersteuning.....</i>	62
<i>Figuur 62: SLES10 Service Pack1 installatie onder Xen (via YaST); let op de ‘x’ voor de apparaten</i>	66
<i>Figuur 63: Windows Server 2008: Server manager</i>	70
<i>Figuur 64: Windows Server 2008: Voeg rol “Hyper-V” toe</i>	71
<i>Figuur 65: Windows Server 2008:Hyper-V update naar Release Candidate 0.....</i>	71
<i>Figuur 66: Hyper-V Manager op Vista</i>	72
<i>Figuur 67: Hyper-V: het aanmaken van een virtuele machine</i>	73
<i>Figuur 68: Hyper-V: alle opties op een rijtje</i>	74
<i>Figuur 69: Hyper-V: de Virtual Network Manager</i>	75
<i>Figuur 70: Hyper-V: Virtual Machine Console</i>	76
<i>Figuur 71: Hyper-V: Virtual Machine Settings</i>	77
<i>Figuur 72: Hyper-V: een fysieke schijf toevoegen</i>	78
<i>Figuur 73: Hyper-V Integration Services</i>	79
<i>Figuur 74: Hyper-V: Apparaatbeheer van geparavirtualiseerde SLES10</i>	80
<i>Figuur 75: Worst Case Test-scenario</i>	82
<i>Figuur 76: Real World Tests scenario</i>	83
<i>Figuur 77: SLES10 installatie: we veranderen het bestandssysteem van reiserfs naar ext3 ..</i>	86

<i>Figuur 78: SLES10: registratie van het systeem waarna we kunnen Online Updaten</i>	87
<i>Figuur 79: Native test: 8 schijven, 4 schijven of intern geheugen</i>	90
<i>Figuur 80: RAID-test: Disk & Read Cache</i>	91
<i>Figuur 81: Hyper-V test met Integration Tools</i>	91
<i>Figuur 82: Barcelona-tests: 1 virtuele machine</i>	92
<i>Figuur 83: Barcelona-tests: 4 virtuele machines</i>	93
<i>Figuur 84: Hyper-V resultaten van fysieke en virtuele schijven</i>	94
<i>Figuur 85: ESX resultaten van fysieke en virtuele schijven</i>	94
<i>Figuur 86: Harpertown-test: GPT of MBR</i>	95
<i>Figuur 87: Server Battle: 3 machines en 3 platformen naast elkaar</i>	96
<i>Figuur 88: Real World scenario: E7330 vs E7350</i>	97
<i>Figuur 89: Real World Scenario: Xen vs VMware</i>	98
<i>Figuur 90: Real World Scenario: E7330 vs Opteron 8356</i>	99
<i>Figuur 91: De D-Link DGS-1224T, een layer-2 gigabit switch met 24 poorten.</i>	106
<i>Figuur 92: Intel SSR212MC2, een storage server</i>	107
<i>Figuur 93: De Promise VTrak M500i, een storage appliance</i>	108
<i>Figuur 94: Een aantrekkelijk GUI zorgt ervoor dat VMware gebruikers zich meteen thuis voelen.</i>	115
<i>Figuur 95: Eenvoudige conversie wizard van Acronis</i>	116
<i>Figuur 96: Omzetten van alle fysieke & virtuele formaten naar alle virtuele formaten.</i>	117
<i>Figuur 97: Intuïtieve GUI van vConverter</i>	118
<i>Figuur 98: SFF-8087 Multilane Cable</i>	A
<i>Figuur 99: SFF-8470 extern miniSAS protocol</i>	A
<i>Figuur 100: SFF-8088 extern miniSAS protocol</i>	B
<i>Figuur 101: SFF-8470 (Enclosure) naar SFF-8088 (Server)</i>	B
<i>Figuur 102: SFF-8484 (backplane) naar SFF-8087 (controller)</i>	B
<i>Figuur 103: SFF-8484 (backplane) naar SFF-8482 (apparaat)</i>	B
<i>Figuur 104: Open-e iSCSI Target Module, voor op een USB-poort zoals rechts afgebeeld.</i> ...	C
<i>Figuur 105: Open-e: 2 laadschermen</i>	D
<i>Figuur 106: Open-e het startscherm met een IP-adres melding</i>	D
<i>Figuur 107: Via de lokale console kunnen we het Open-e IP-adres instellen</i>	E
<i>Figuur 108: Open-e startvenster na inloggen via een webbrowser</i>	F
<i>Figuur 109: Open-e volume manager</i>	F
<i>Figuur 110: Open-e: Volume groups</i>	G
<i>Figuur 111: Open-e: aanmaken van een iSCSI Target</i>	H
<i>Figuur 112: Open-e: volume toewijzen aan een target</i>	H
<i>Figuur 113: Promise VTrak 15200 Storage Appliance</i>	I
<i>Figuur 114: Promise VTrak M500i Storage Appliance</i>	J
<i>Figuur 115: HP MSA1510i iSCSI Controller</i>	K
<i>Figuur 116: Hardware overzicht: de Intel SSR212MC2</i>	L
<i>Figuur 117: Hardware overzicht: VTrak E310f bovenaan, Intel SRR212MC2 onderaan</i>	M
<i>Figuur 118: Hardware overzicht: Promise VTrak J300S</i>	M

<i>Figuur 119: Hardware overzicht: Barcelona, Harpertown & Clovertown in hetzelfde chassis</i>	N
<i>Figuur 120: Systeem eigenschappen van Windows Server 2008</i>	R
<i>Figuur 121: Windows Server 2008 Powershell</i>	T
<i>Figuur 122: Windows Server 2008 Internet Information Services 7</i>	U
<i>Figuur 123: Terminal Services in Windows 2008: flowchart</i>	V

TABELLIJST

<i>Tabel 1: Tabel met CPUs die ondersteuning bieden voor hardware virtualisatie.</i>	44
<i>Tabel 2: Systeemeisen voor Windows Server 2008</i>	69
<i>Tabel 3: Overzicht Quad Core Processors</i>	97
<i>Tabel 4: ESX versies en hun ondersteuning.</i>	104

1. Inleiding

De wereld is een achtertuin. Door de technologische vooruitgang en dankzij het internet is iedereen altijd online. Er zijn ook enorm veel bedrijven en KMO's die globaal werken. Zij leveren diensten en applicaties af over de volledige wereld. Ze willen niet dat hun website uitvalt voor de (potentieel) grote Amerikaanse klanten. Wat is hierbij het grootste probleem? Hoe zorg je er namelijk voor dat je applicatie (website, database, beheersysteem ...) 365 dagen op 365 bereikbaar is? Hoe zorg je voor de zogenaamde High Availability?

Het logische antwoord is hier: Redundantie (redundancy). Door 2 servers tegelijk op te zetten met dezelfde applicatie kan de ene overnemen als de andere faalt. Maar de meeste bedrijven (en dan vooral IT bedrijven) houden het niet bij 1 of 2 applicaties en dus bij 1 of 2 servers. Ze hebben er al snel een tiental. Dit betekent op HA-niveau meteen 20 servers, wat een probleem levert naar kostprijs (TCO) en milieuvriendelijkheid. Hoe lossen we dit op?

Antwoord: Virtualisatie.

Virtualisatie is de techniek om meerdere besturingssystemen tegelijk en onafhankelijk op eenzelfde server te draaien of om meerdere applicaties tegelijk en apart te draaien op eenzelfde server. Je zou anders kunnen zeggen dat virtualisatie het parallel draaien is van applicaties of besturingssystemen op één en dezelfde hardware.

Parallel draaien wil zeggen dat de hardware gedeeld wordt over verschillende applicaties en besturingssystemen en daar zit uiteraard het grootste probleem. Wat is het gevolg voor de responstijd & algemene throughput van het virtualiseren van applicaties? In hoeverre kan een, aan virtualisatie aangepaste, hardware helpen bij deze virtualisaties? Daar probeert dit project een antwoord op te bieden.

2. Shared Storage

2.1. Inleiding

De opdracht was hier het specifieke onderzoek naar de mogelijkheden voor SAN-opstellingen. Bepalend hierbij was de prestatiefactoren van iSCSI zodat deze vergelijkbaar wordt met het alternatief, zijnde Fibre Channel. Met behulp van enkele tests en instellingen, die toepasbaar zijn binnen de reële bedrijfswereld, worden deze dan geschetst in laatste onderdeel van het shared storage hoofdstuk.

2.1.1. Storage Hardware, kennismaking met SAS¹

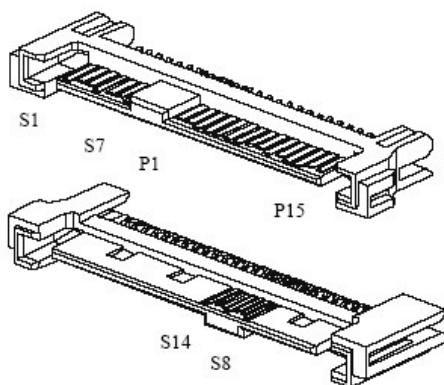
Al enkele jaren bestaat er in de serverwereld een zeer deftige opvolger voor het oude SCSI-verhaal (wat eindigde met de Ultra SCSI 320). Deze opvolger heet 'SAS', wat staat voor Serial-Attached-SCSI. Dit komt neer op het versturen van SCSI-commando's over een S-ATA interface.

SAS heeft ook standaard de terminator ingebouwd, wat het weer dichterbij S-ATA laat aanleunen.

In de SAS standaard zit echter nog iets extra's, het is voorzien om in "Dual-path" of "Dual-mode" te werken. Dit is een soort redundantie, maar dan op het fysieke niveau van de schijf zelf. Dit betekent dat er 2 datakabels per schijf voorzien zijn, zodat een uitgetrokken of kapotte kabel niet resulteert in een niet-werkende schijf.

In deze standaard zit ook NCQ verwerkt (Native-Command-Queuing), wat voor een hoger rendement zorgt.

Een voordeel van een SAS-systeem is dat het zeer eenvoudig uit te breiden is, wat een hoge betrouwbaarheid genereert. Dit maakt van SAS de ideale basis voor SAN.



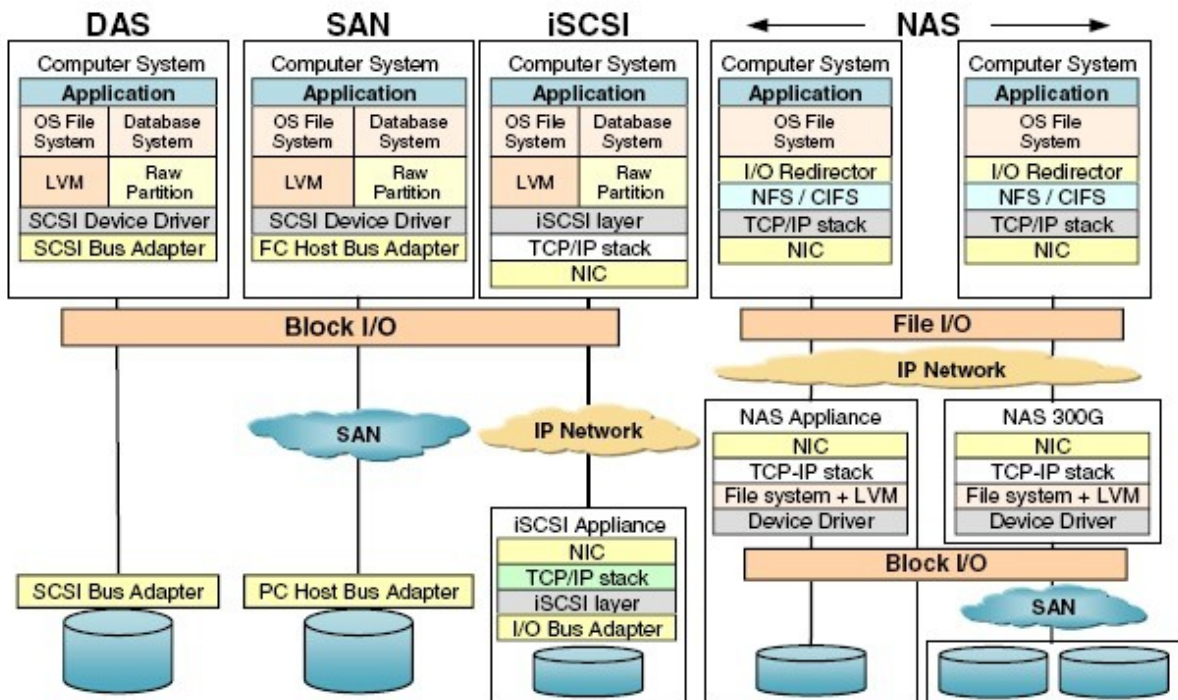
Figuur 1: Fysiek verschil tussen SAS & SATA schijfinterface



Figuur 2: Het is mogelijk 2 systemen een SAS loopwerk te laten gebruiken, in theorie is zelfs dubbele doorvoersnelheid mogelijk

¹ Appendix A: SATA/SAS bekabelings Overzicht

2.1.2. DAS, NAS & SAN



Block I/O compared to file I/O

Storage characteristic	iSCSI SAN	Fibre Channel SAN	NAS
Protocol	Serial SCSI	Fibre Channel Protocol	NFS, CIFS
Network	Ethernet, TCP/IP	Fibre Channel	Ethernet, TCP/IP
Source / target	Server / Device	Server / Device	Client / Server or Server / Server
Transfer	Blocks	Blocks	Files
Storage device connection	Direct on network	Direct on network	I/O bus
Embedded file system	No	No	Yes

Figuur 3: Schema met overzicht van alle verschillende opslagtechnologieën

Om een server toegang te verlenen tot opslag zijn er 3 types:

▲ Direct Attached Storage:

Dit is de meeste voorkomende opslagmogelijkheid bij werkstations en laptops. Dit omvat lokale harde schijven, USB Sticks, Cd's en dergelijke. Enkel de eigen machine heeft rechtstreeks toegang tot deze opslag.

▲ Network Attached Storage:

NAS omvat alle opslag die over het netwerk gedeeld wordt. Hier worden bestanden doorgestuurd over het TCP-IP netwerk en geen blokken. Aan de server kant hebben we dus een (minimaal) OS dat rechtstreekse toegang heeft tot de opslag en bestanden dan op zijn beurt over het netwerk doorstuurt naar de aanvrager. Dit netwerk is een standaard Ethernet/TCP-IP-netwerk zoals het internet.

▲ Storage Area Network:

Hier heeft de aanvrager, over een netwerk, rechtstreeks toegang tot de adapter die de schijven aanspreekt. Het zijn niet de bestanden die verstuurd worden, maar wel datablokken. Het opslagapparaat is rechtstreeks aangesloten op een storage netwerk. Dit netwerk kan een Fibre Channel Network zijn, maar ook een Ethernet/TCP-IP netwerk, afhankelijk van het gebruikte protocol. Het voordeel van SAN is dat het cliënt OS niet ziet dat het om netwerkopslag gaat, waardoor bestanden gedeeltelijk bewerkt kunnen worden i.p.v. volledig door te sturen. Op die manier kunnen complexe databasesystemen beheerd worden via SAN.

Van de 3 types opslagmogelijkheden is voor het project enkel SAN van nut. KMO's doen forse investeringen in SAN opstellingen en als we deze voldoende kunnen optimaliseren kunnen we deze KMO's heel wat kosten besparen.

2.2. SAN: Fibre Channel of iSCSI

Er zijn 2 mogelijkheden voor een Storage Area Network: Fibre Channel en iSCSI.

Fibre Channel is zeer duur en een aparte kennis is nodig om dit soort gespecialiseerde netwerken aan te leggen.

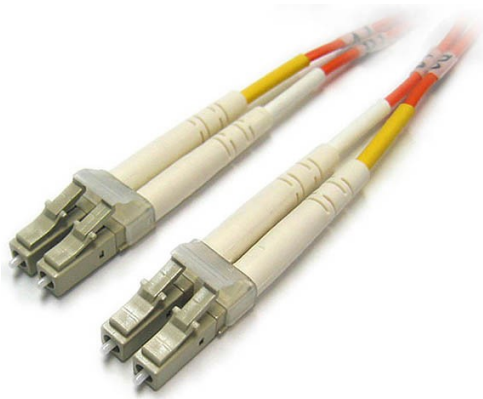
De iSCSI standaard ligt nog maar vast sinds 2002 en vanaf 2005 begonnen de gespecialiseerde apparaten door te sijpelen. Het stuurt de block aanvragen over een standaard Ethernet TCP-IP netwerk en kan dus gewoon met (gigabit) switches gebruikt worden, wat het niet alleen goedkoper maakt, maar ook flexibeler.

2.2.1. Fibre Channel

Fibre Channel werkt met kostbare & breekbare glasvezelkabels. Het vereist aparte, dure Fibre Channel switchen en in iedere cliënt dient een FC HBA (Host Bus Adapter of insteekkaart) te zijn ingeplugd.



Figuur 4: Fibre Channel Switch (LightSand S-8100A) ≈ \$ 5,0000.00



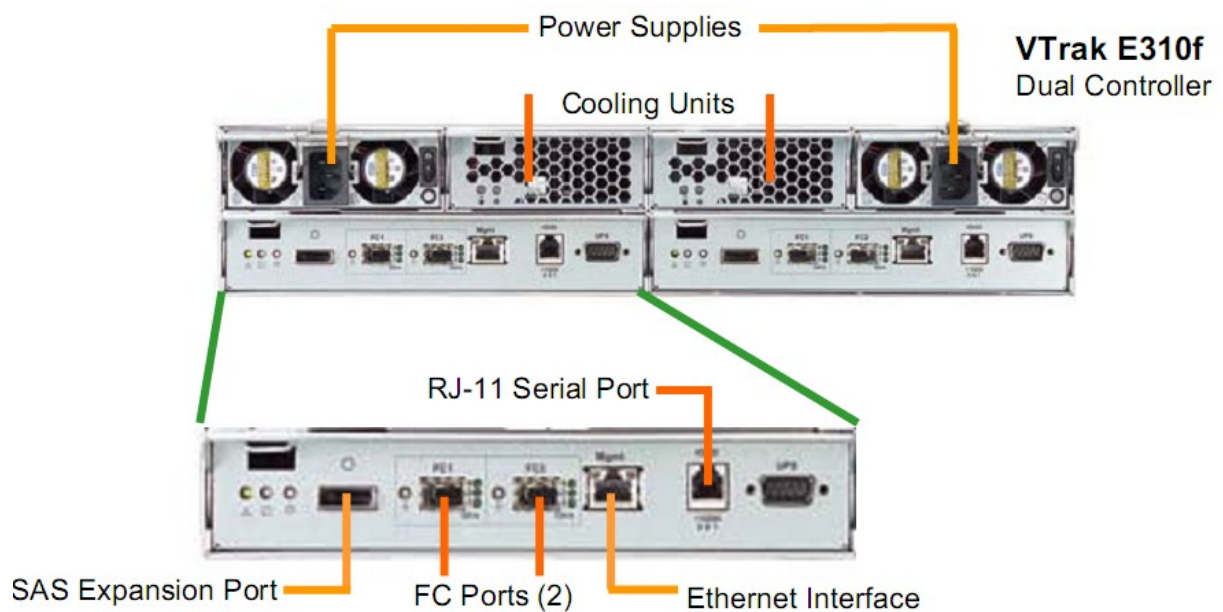
*Figuur 5: Glasvezel LC-LC Kabel
Duplex, 1 zendkanaal en 1 ontvangkanaal*



*Figuur 6: Fibre Channel Host Bus Adapter
PCI-Express 4X*

Fibre Channel bestaat in 4 snelheden: 1, 2, 4 of 8Gbit/sec. 10Gbit/sec is nog in ontwikkeling en zal niet backwards compatible zijn.

Er is ook een Fibre Channel opslag apparaat nodig, zoals de Promise E310F die we gebruikten tijdens het project. Hier kunnen 2 controllers inzitten die elk toegang hebben tot de schijven-array. Hij kan tevens 12 schijven bevatten die zowel SATA als SAS kan zijn. Er zit dus ingebouwde logica in, die RAID opstellingen mogelijk maakt. Iedere controller bevat twee 4Gbit FC-poorten. De ethernet en seriële interface dienen om het geheel op afstand te beheren. En de SAS Expansion poort kan gebruikt worden om meerdere (JBOD-)oplossingen met elkaar te verbinden, dit heet Daisy Chain.



Figuur 7: Promise vTrak E310f Fibre Channel Storage Array

2.2.2. iSCSI

2.2.2.1. iSCSI's specifieke terminologieën:

- ▲ Een target draait op de server (als service) en voorziet communicatie tussen de initiator en de onderliggende schijven of partities.
- ▲ Een initiator draait op de cliënt (als service) en vangt de iSCSI signalen, komende van het target, op en geeft deze door naar het onderliggende OS.
- ▲ LUN staat voor Logical Unit Number en heeft, naargelang de toepassing, meerdere interpretaties. Bij iSCSI is dit meestal een schijf of partitie die wordt gekoppeld aan één of meerdere targets.
- ▲ iQN (iSCSI Qualified Name): dit is een unieke naam die zowel target als initiator altijd krijgen, en hierop wordt de eventuele authenticatie gedaan. Ze wordt als volgt samengesteld:
iqn.yyyy-mm.{omgekeerde domeinnaam}:naam (bijv: iqn.1995-01.com.microsoft:test-target1)

2.2.2.2. iSCSI implementatie

Er zijn 3 manieren om iSCSI in een KMO te implementeren:

- ▲ Server met iSCSI Target:
dit is een standaard server, kan zelfs toren of workstation zijn, waar via DAS enkele schijven of een array van schijven aanhangen. Deze kunnen dan via een software toepassing ingesteld worden als LUNs voor één of meerdere targets. Deze LUNs kunnen gewoon naast het OS staan, zelfs op dezelfde schijf.
Hiervoor is de meeste kennis vereist, want alles moeten we nog zelf installeren. Er bestaan ook oplossingen op een USB Stick zoals *Open-E*².
- ▲ Storage Server met iSCSI Target:
dit zijn al gespecialiseerde servers. Ze bestaan uit 2 delen, vooraan de machine zit een storage rack, met ingebouwde RAID controller. Achter die storage rack zit een redelijk standaard server, die nog eens eigen systeemschijven heeft. Het opslag gedeelte zit dus volledig gescheiden van het server gedeelte. Deze zijn verkrijgbaar als volledige oplossingen, met het target software al geïnstalleerd (bijv. HP). Maar we kunnen de storage server ook apart kopen, zoals de hier gebruikte Intel SSR212MC2:

² Appendix B: Eerste ervaringen met Open-E



Figuur 8: Intel SSR212MC2

▲ Storage Appliance³:

Dit is het meest gebruiksvriendelijk voor de bedrijven, gewoon kopen en inpluggen op het netwerk. Via een management interface kunnen we dan de verschillende LUNs en targets aanmaken. Het OS is meestal gestript en aangepast voor iSCSI.

2.2.2.3. iSCSI software fabrikanten

Er zijn verschillende fabrikanten van initiators en targets. In theorie zouden deze onderling met elkaar moeten kunnen werken, maar het is logischer om dezelfde fabrikanten te gebruiken.

Initiator mogelijkheden:

- ▲ Microsoft Initiator; is gratis te downloaden en te gebruiken. Het werkt op nagenoeg ieder Windows OS en is vanaf Vista zelfs standaard ingebouwd.
- ▲ Linux Initiator; deze is (uiteraard) ook gratis en zit o.a. standaard ingebouwd in SLES 10 (SUSE Linux Enterprise Server).
- ▲ VMWare iSCSI Initiator; deze is een initiator die in ESX zit, waardoor deze software ook via iSCSI kan werken en is sterk gebaseerd op de Linux Initiator.
- ▲ Rocket Division StarPort; deze heeft een simpele interface en werkt op dezelfde manier als de MS initiator.

Target mogelijkheden:

- ▲ Linux iSCSI Enterprise Target; standaard ingebouwd in SLES 10 en wellicht ook door HP gebruikt in zijn appliances en storage servers. Er is ook een aparte oplossing zoals Open-E die deze target gebruikt.
- ▲ Microsoft iSCSI Software Target; deze wordt enkel via OEM geleverd en is dus onvindbaar op de markt, hij kan voorgeïnstalleerd worden op de MS Windows 2003 Storage Server.

³ Appendix C: Enkele voorbeelden van Storage Appliances

- ▲ MySAN; is een mindere speler op de markt want hij laat slechts toegang van 2 servers toe.
- ▲ Rocket Division StarWind; dit is een Windows iSCSI target met zeer veel mogelijkheden, maar niet erg gebruiksvriendelijk.

2.2.2.4. Target Accessmodes

Een target & een initiator kunnen op verschillende manieren communiceren met elkaar. Deze werden ons duidelijk gemaakt door Anton A. Kolomyeytsev van Rocket Division en we hebben deze door eigen tests onderzocht (zie later). Bij de StarWind Target zijn al deze accesmodes instelbaar, bij andere is er de keuze gemaakt door de fabrikant.

- ▲ SPTI (SCSI Pass Through Interface):
ontwikkeld door Microsoft. Andere alternatieven zijn SPTD van Duplex Secure (gebruikt door Daemon Tools) of ASPI van Adaptec. Bij het gebruik van deze mode worden alle commando's via het target doorgegeven aan Windows, meerbepaald aan de SPTI driver. Deze driver spreekt dan op zijn beurt het opslag apparaat aan. Voordeel van deze access mode is dat nagenoeg ieder apparaat, dat werkt onder Windows, kan gedeeld worden via iSCSI. Zo ook Blu-Ray branders, tape-devices, ... Het grootste nadeel is dat het traag werkt. (zie testresultaten)
- ▲ Disk Bridge:
met deze methode wordt rechtstreeks de volledige harde schijf gebruikt, zoals ze zichtbaar is onder Windows. Dus inclusief alle partities. De server ziet deze partities dus ook waardoor "vervuiling" door het server OS mogelijk is. Tevens worden nog alle iSCSI signalen geëmuleerd, waardoor alle harde schijven & HBA-configuraties mogelijk zijn. Er is echter geen non-serialized access mogelijk. (zie: prestatiefactoren)
- ▲ Image File Mode:
is de meest performante mode en wordt ook gebruikt door MS iSCSI Software Target. Hier wordt per bestand één LUN gemaakt bovenop het NTFS bestandssysteem van Windows, dit bestand is dan de harde schijf die wordt getoond aan de initiator. Het voordeel hiervan is dat Windows geen toegang heeft tot dit bestand en dat het target deze bestanden tot op de sector kan aanspreken. Ook caching is hier mogelijk. Dit is de werkwijze waarop, bijvoorbeeld, de MS iSCSI Software Target werkt.

2.3. Prestatiefactoren

Er zijn meerdere niveaus waarop we optimalisaties kunnen toepassen. Er zijn de verschillende access modes die zonet besproken werden, maar er kan ook standaard netwerk-tuning worden uitgevoerd (auto-negotiation, TOE, VLANs, Jumbo Frames). Op de fysieke hardware kunnen we ook veel aanpassen, zowel op de RAID Controller (caching) als op de harde schijf en het OS (partition offsets, clustersizes).

iSCSI Tuning factoren

- ▲ Serialized <-> non-serialized access = Untagged <-> Tagged command queuing:
TCQ wil zeggen dat het target tags kan meesturen, waardoor ze out-of-order kunnen afgehandeld worden. De controller kiest dan zelf om eerst de pakketjes te lezen die zich dicht bij elkaar bevinden op de schijf, zodat de lees/schrijf-kop minder moet bewegen. Dit heeft een grote prestatiewinst tot gevolg.
Untagged (=serialized) sending maakt de pakketjes weliswaar kleiner, maar ze moeten wel uitgevoerd worden in de volgorde dat ze aankomen. De lees/schrijf-kop moet alle bewegingen dus onmiddellijk uitvoeren.
- ▲ Multi-Path I/O (MPIO):
Dit bevindt zich meer op het iSCSI protocol niveau, het is de mogelijkheid om met meerdere netwerkkaarten via verschillende manieren in te loggen op het iSCSI Target. Niet alleen kan hiermee de bandbreedte verhoogd worden, maar tevens kunnen we in Fail-Over werken. Als dan 1 netwerkkaart, of het netwerk waaraan deze hangt, plat gaat neemt de andere automatisch over. Dit werkt zowel voor target als de initiator.

Netwerk-tuning

- ▲ Jumbo Frames:
ofwel TCP Stack Tuning, dit is het groter maken van de Ethernet pakketjes, bijv. van 1500 naar 9000 bytes of meer. Dit zorgt uiteraard dat er per hoeveelheid data maar 1 header/footer/... moet zijn en dus meer data per keer kan verstuurd worden. Bijgevolg is dit een performance boost voor grote hoeveelheden data.
- ▲ Auto-negotiation off:
dit is het proces waarbij 2 netwerkkaarten (of Netwerkkaart-switch) onder elkaar de beste communicatieparameters, zoals snelheid & duplex-mode, bepalen. Maar voor iSCSI wordt deze auto-negotiation best uitgeschakeld en alles manueel op maximum gezet, waardoor Jumbo Frames ook meteen meer voordeel halen.
- ▲ TOE / TCP Offload Engine:
Dit is een technologie die moet ondersteund worden door de netwerkkaart in kwestie. Het komt erop neer dat de TCP/IP berekeningen afgeleid worden van de CPU naar de TOE Chip op de NIC. Dit is vooral handig bij minder snelle CPU's. Dikwijls hebben TOE kaarten en andere hardware apparaten eigen initiator firmware, waardoor initiator software niet langer nodig is. Maar deze kaarten zijn niet enorm populair wegens hoge kostprijs en feit dat recente CPU's snel genoeg zijn. We vinden ze bij verkopers als Alacritech, LeWiz Communications Inc en QLogic Corp.
- ▲ VLAN samen met Layer 3 Switchen:
Layer 3 duidt in eerste instantie op de laag van het OSI model waarop de switch de pakketjes bekijkt, layer 2 is frame, layer 3 is pakket.
Een layer 3 switch is meestal een intelligente switch die o.a. VLAN ondersteuning heeft. Een VLAN is een Virtueel Netwerk dat volledig los staat van de andere netwerken. Dit VLAN heeft er geen last van, en er is dus ook geen overhead (van bijvoorbeeld broadcasts). Het data & TCP/IP verkeer zal het verkeer op het andere netwerk niet beïnvloeden. Zoals later zal blijken, is deze meer een veiligheidsinstelling dan dat er veel prestatiewinst zal gegenereerd worden.

Fysiek (RAID controller & OS)

▲ Caching:

Schrijfcache is er dikwijls op het niveau van de Raidcontroller (write-back), indien er een aparte batterijmodule is voor de Raidcontroller. Deze batterijmodule is nodig om zeker te zijn dat er (in het geval van stroomuitval) geen data verloren gaat.

Sommige targets kunnen caching voorzien bij het lezen ook. Ze doen dat door de gelezen data op te slaan in het lokaal werkgeheugen van de server, dit is veel sneller dan de harde schijven. Ook StarWind zou deze caching ondersteunen, maar voor MS iSCSI Software Target is dit nog niet het geval. Dit betekent ook dat de grootte van het geheugen belangrijk is voor de caching, dit wordt aangeduid bij één van de testen.

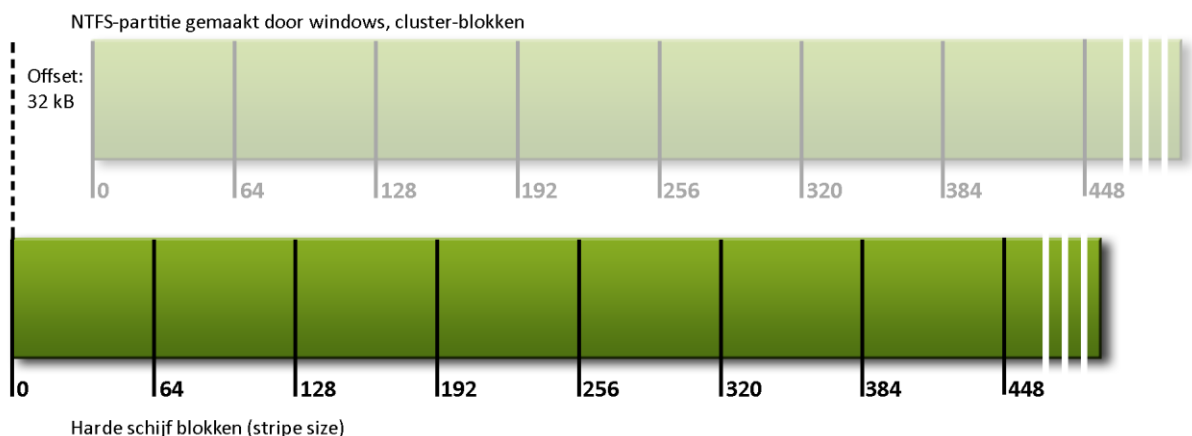
▲ Partition offset:

Een fysieke schijf wordt ingedeeld in stripe blokken (vooral bij Raidopstellingen), deze blokken zijn meestal 64 kB groot. Windows maakt zijn (NTFS-)partities standaard aan met een offset van 31,5 kB. En tevens met een clustergrootte van 64 kB (gangbaar voor iSCSI partities), wil dit zeggen dat 1 cluster telkens 2 stripes overlapt. Zie schema hieronder voor verduidelijking:

Dit probleem is erkend door Microsoft en besproken in Knowledge Base Q923332

Microsoft maakt primaire systeem partities aan met een offset van 32 kB

Offset 0x3F = sector 63 = 31,5 kB ≈ 32 kB



Figuur 9: Zelfgemaakt schema ter verduidelijking van het partition offset probleem

Dit betekent dat als er 1 cluster nodig is, er telkens 2 requests nodig zijn wat overvloedige bandbreedte oplevert. Het verschil zal hier inderdaad enkel zichtbaar zijn bij willekeurige leesopdrachten.

Vista lost dit op met SP1 en vanaf Windows Server 2008 staan alle offsets standaard op 1024KB.

2.4. Concrete opbouw

De theorie is achter de rug, tijd om zelf aan de slag te gaan, hier een kleine howto over hoe we een iSCSI systeem opbouwen en testen.

We gebruiken hier Microsoft iSCSI Target Software en voor de howto toon ik SQLIO, omdat daar de meeste realistische tests mee kunnen gebeuren.

2.4.1. Microsoft iSCSI Software Target (WinTarget)

2.4.1.1. Inleiding

MS iSCSI Target werkt volledig anders dan de al bekende Target Software (StarWind & Linux). Deze laatste slaagt erin om volledige schijven te delen (starwind & Linux), of volledige partities (Linux), maar MS iSCSI Target werkt met LUNs.

Een LUN is eigenlijk niks anders dan een bestand, een virtuele harde schijf die we aanmaken en meteen een vaste grootte geven.

Deze LUN heeft als voordeel dat we hem on-the-fly kunnen vergroten en verkleinen, maar ook eenvoudig kunt kopiëren (backuppen) of verplaatsen. We kunnen hem tevens mounten in windows zodat we hem kunnen bekijken via de Windows verkenner.

Trouwens deze LUN kan groottes hebben, ver boven 500GB, daar dit overeen komt met een bestand van 500GB dient de fysieke harde schijf als NTFS geformatteerd te zijn.

2.4.1.2. Vereisten (prerequisites)

- ▲ Werkt enkel op Windows Storage Server R2 (Service Pack 2); Enterprise of Standard
- ▲ Storage Manager for SANs moet geïnstalleerd staan (= standaard bij de Storage Server)

2.4.1.3. MS iSCSI Target installeren

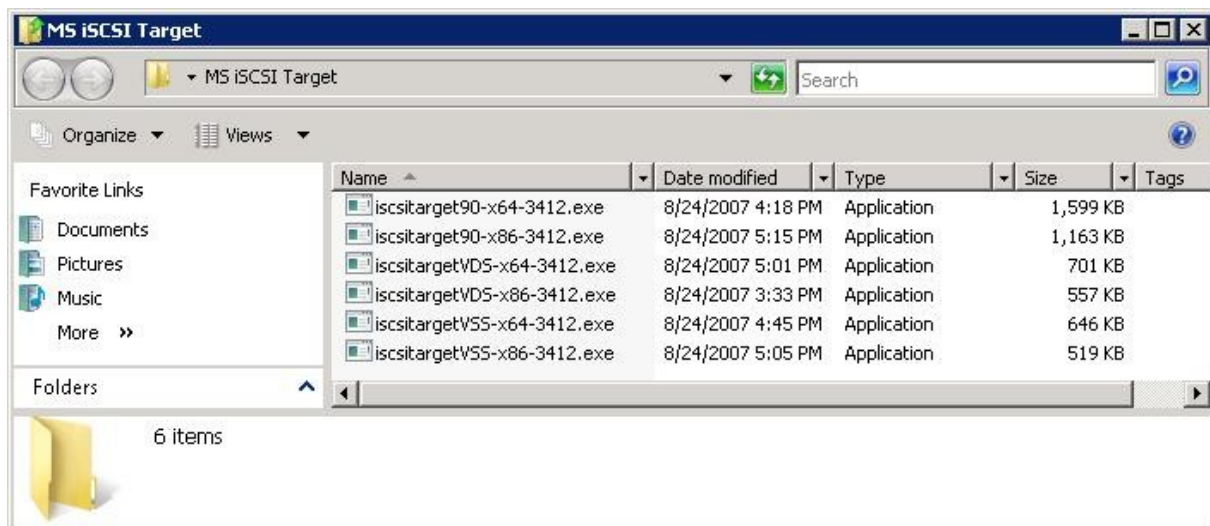
Eerst dient de Virtual Disc Service geïnstalleerd te worden, dit kan met `iscsitarget-vdss-3.0-1983-x86fre.exe`

Het achtervoegsel `fre` of `chk` slaat op de `free` of de `checked` versie. De eerste is sterk te vergelijken met een retail-versie, terwijl de tweede eerder een debug-versie is. Een `checked` heeft dus dikwijls nog `traces` en `asserts` ingebouwd.

Vandaar de `Free`-keuze, als er problemen zijn kan er altijd geopteerd worden voor de `Checked` versie.

VSS staat voor Volume Shadow Copy Service en is interessant om snapshots te maken van virtuele schijven. Als er iets gebeurt, kan er altijd teruggekeerd worden naar deze "herstelpunten".

Daarna dient de iSCSI Target Software zelf nog geïnstalleerd te worden, dit kan met: `iscsitarget-3.0-1983-x86fre.exe`.

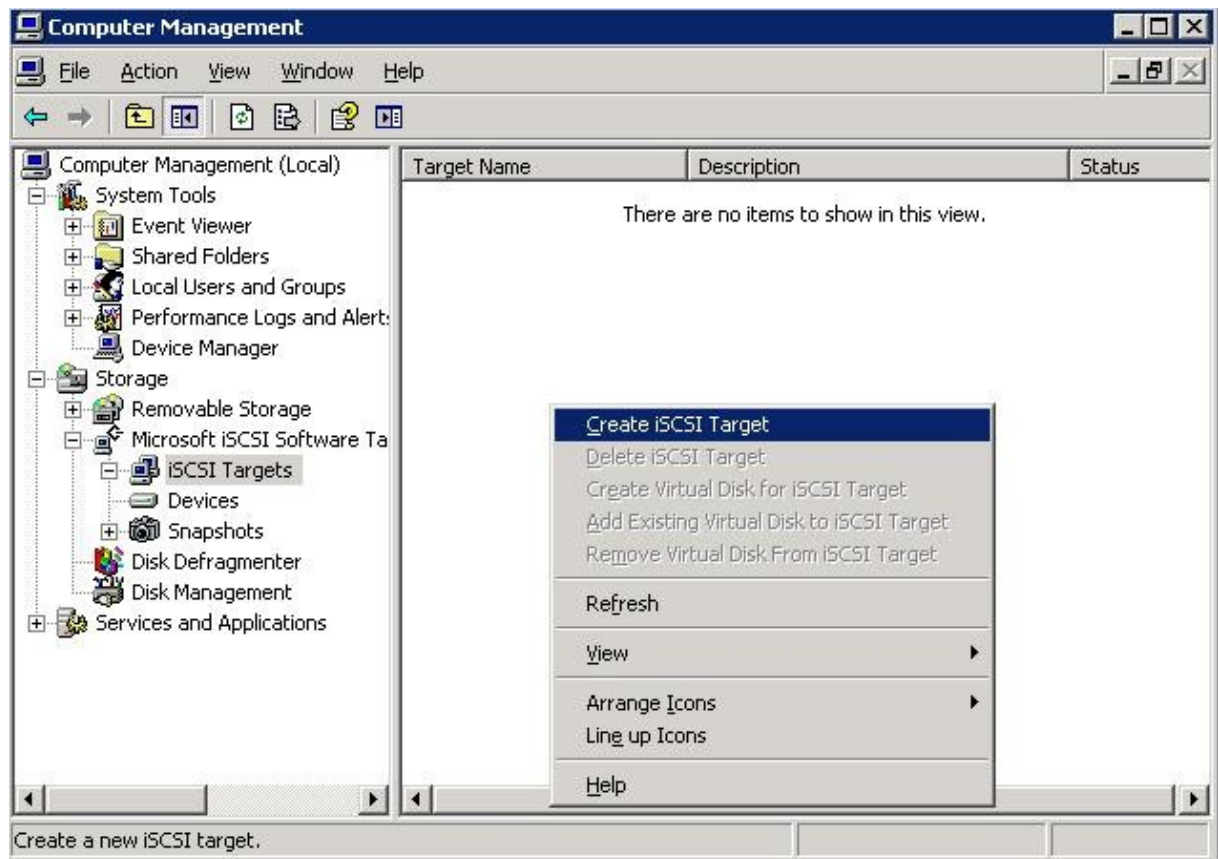


Figuur 10: iSCSI Target Software: Er is een 64bit & een 32bit versie

2.4.1.4. Opzetten van een target

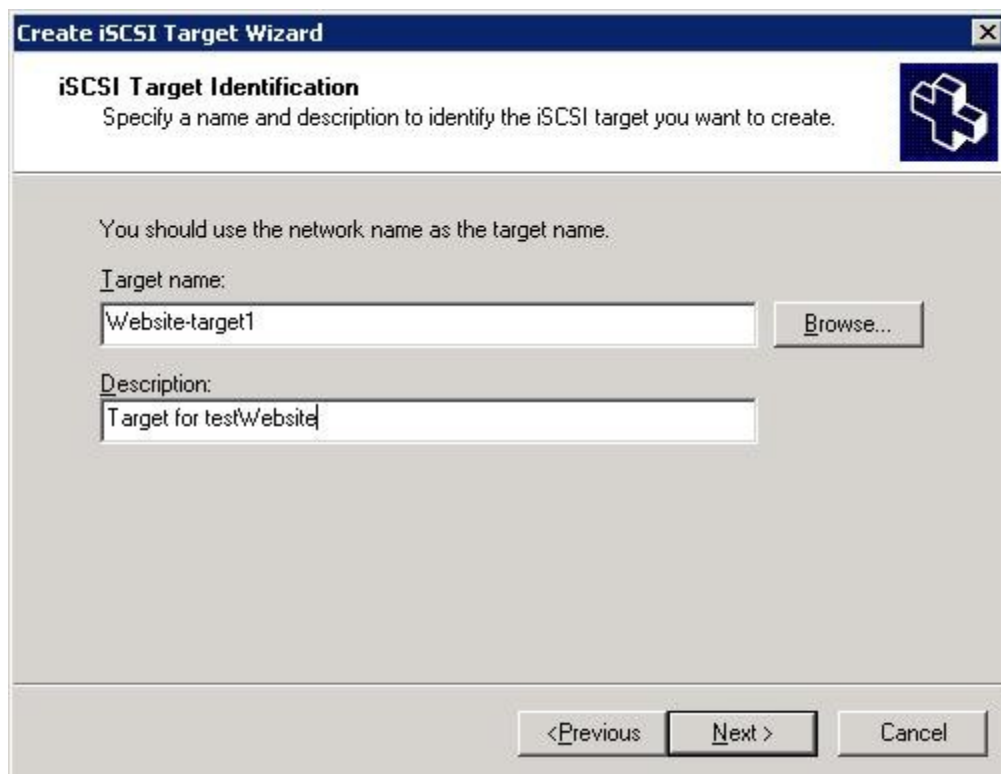
Dit wordt gedaan aan de hand van een voorbeeld, maar dit is zeer eenvoudig door te trekken naar de meeste, zoniet alle, systemen.

- We bekijken eerst het Disk Management, we zien een (lege) RAID-schijf van 271GB die als drive-letter D: heeft. Hierop maken we onze LUN aan zodat we deze schijf kunnen gebruiken/testen.
- Na installatie van de Microsoft iSCSI Target Software is er extra onderdeel zichtbaar in het Computer Management (te openen via Control Panel), genaamd “Microsoft iSCSI Software Target”, dat openklikken waarna een iSCSI Target aan te maken valt.



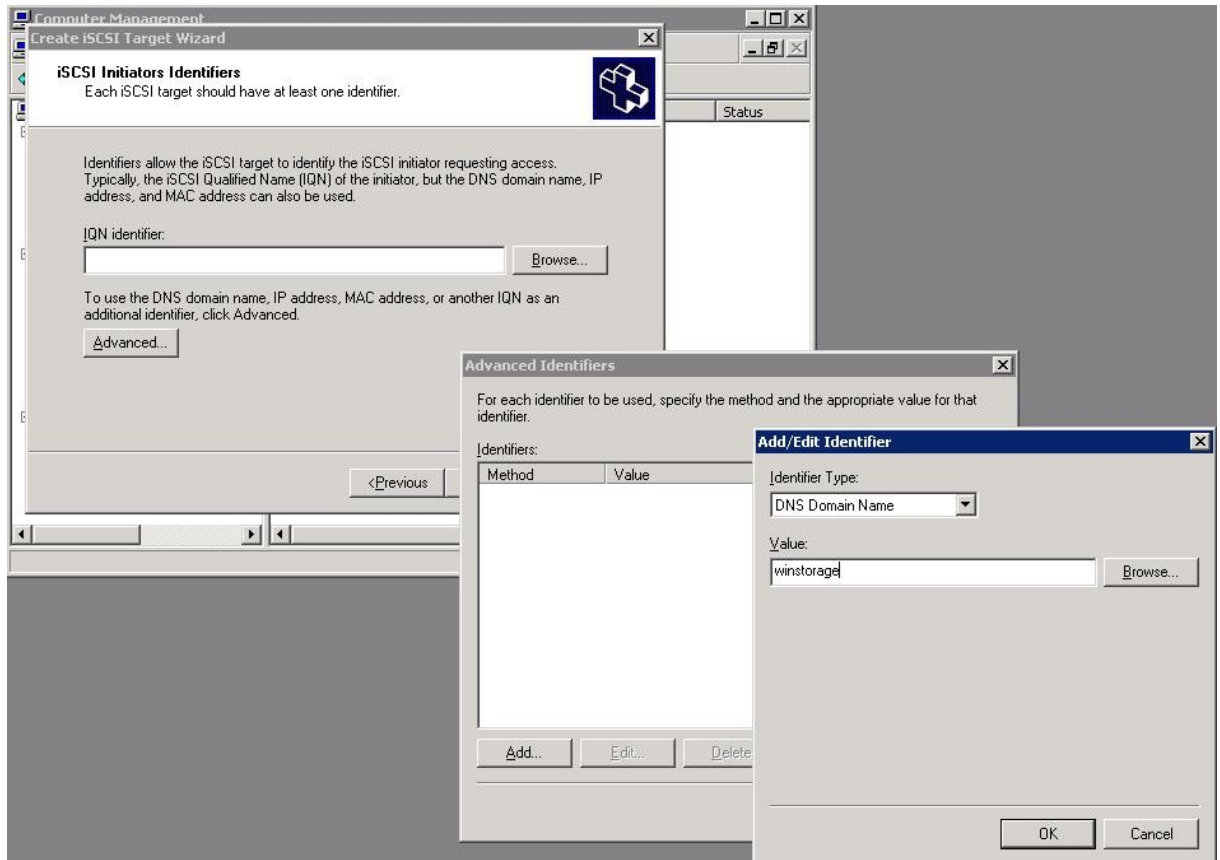
Figuur 11: Opzetten iSCSI Target

- Dan wordt een wizard geopend waarin een verplichte naam en een optionele beschrijving kunnen worden getypt.



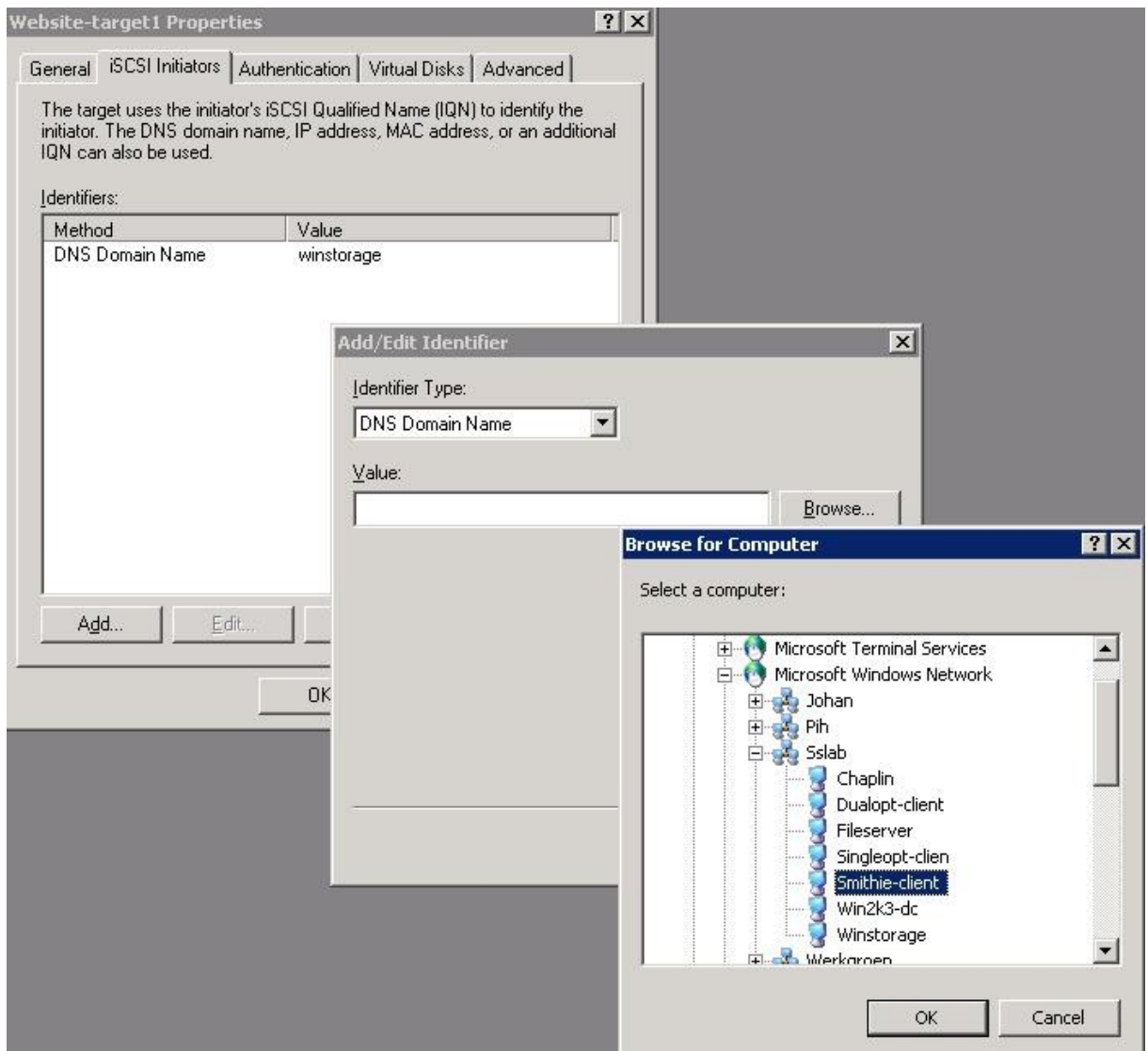
Figuur 12: Naamgeving iSCSI Target

- Na een druk op “Next” wordt er gevraagd achter een unieke IQN, hier zijn meerdere mogelijkheden. Aangezien we eerder gezien hebben hoe een IQN opgebouwd is kunnen we er één verzinnen, maar eenvoudiger is klikken op “Advanced...” waarna we met een druk op “Add...” kunnen kiezen voor *IQN*, *DNS Domain Name*, *IP Address* of *Mac Address*. We geven hier onze domeinnaam in. Terug 2 maal OK en dan een druk op “Next” en “Finish”.



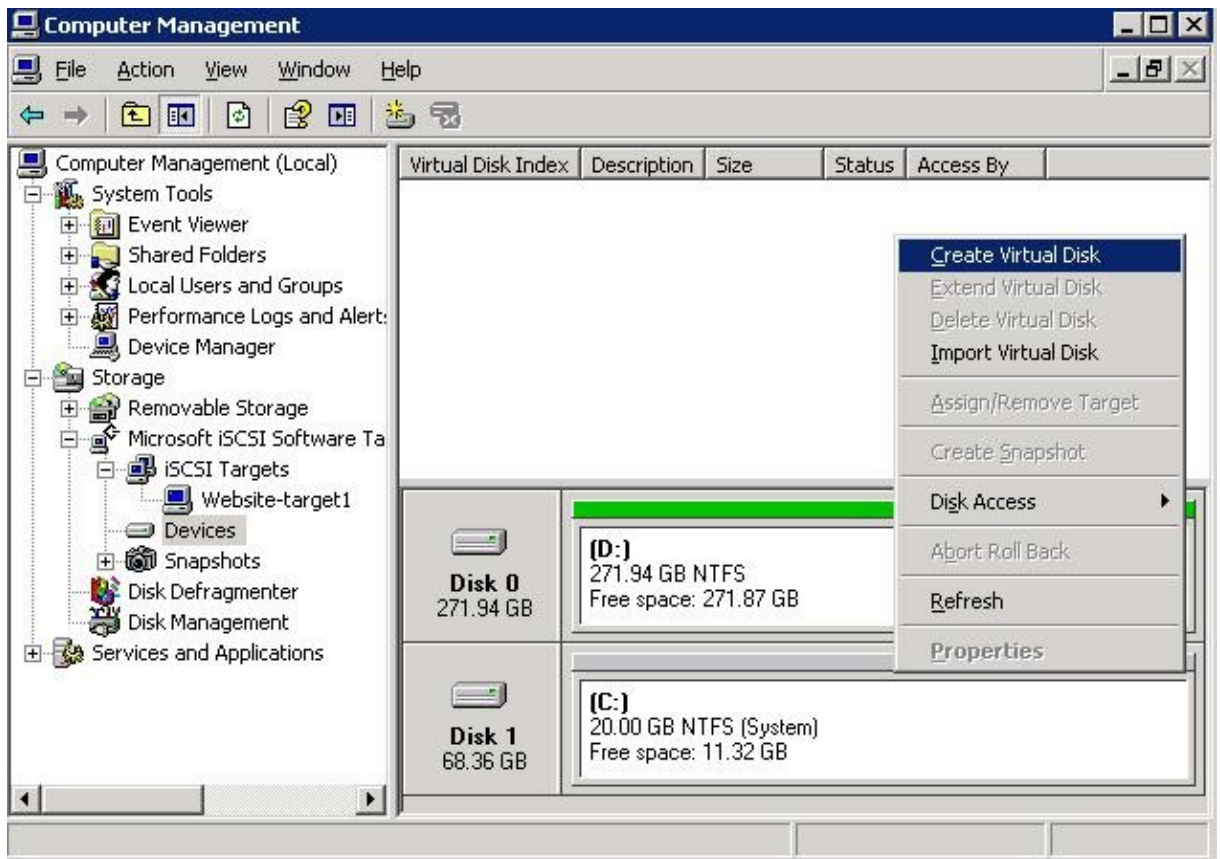
Figuur 13: iSCSI initiator toevoegen aan target

- Het target zelf is nu wel aangemaakt, maar merk op dat we er niet meteen een schijf aan kunnen verbinden (ook geen virtuele). Dit dienen we achteraf te doen (eventueel bij het aanmaken van de virtuele schijf). Wat we ook nog moeten doen is een initiator toevoegen aan de lijst van goedgekeurde initiators bij het target.
- We gaan hiervoor naar de properties van het target en dan het tabblad “iSCSI Initiators”. We klikken “Add...”, dan terug “DNS Domain Name”. We kunnen natuurlijk ook browsen naar de juiste client.



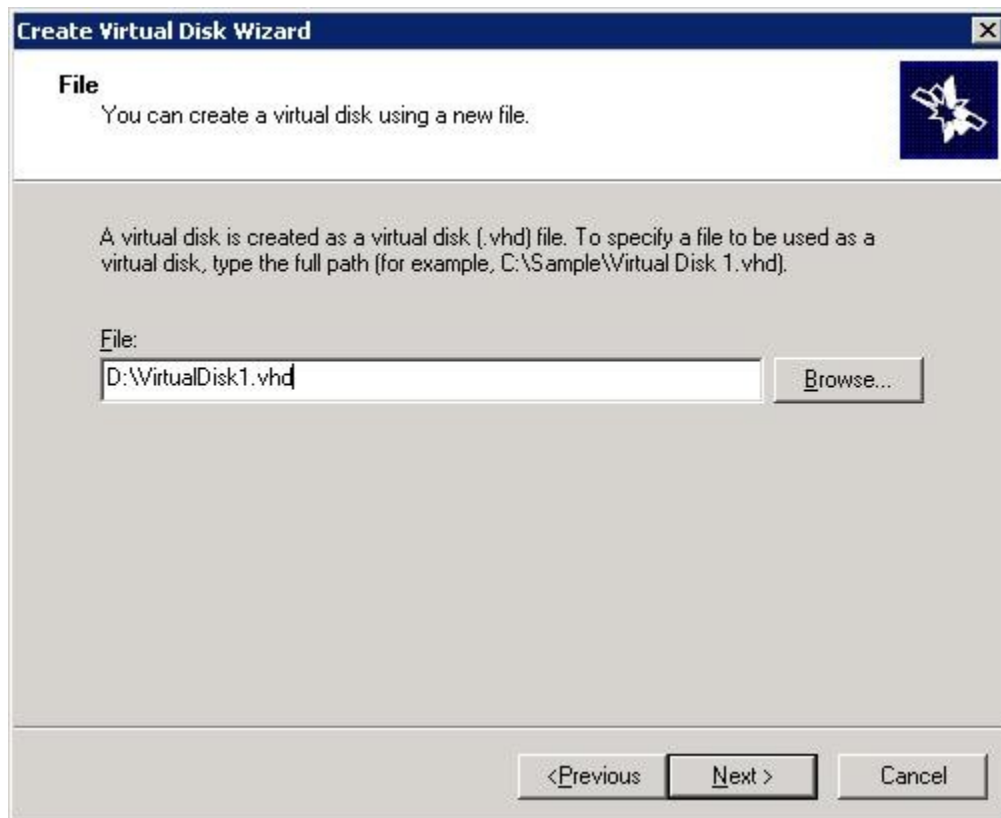
Figuur 14: iSCSI Initiator, DNS Naam kiezen

- Nu is het target volledig en kan de client zelfs al verbinden, maar een schijf zal deze cliënt niet vinden.
- We maken nu een virtuele schijf aan; hiervoor gaan we naar de optie “Devices” in het Computer Management en kiezen voor Create Virtual Disk:



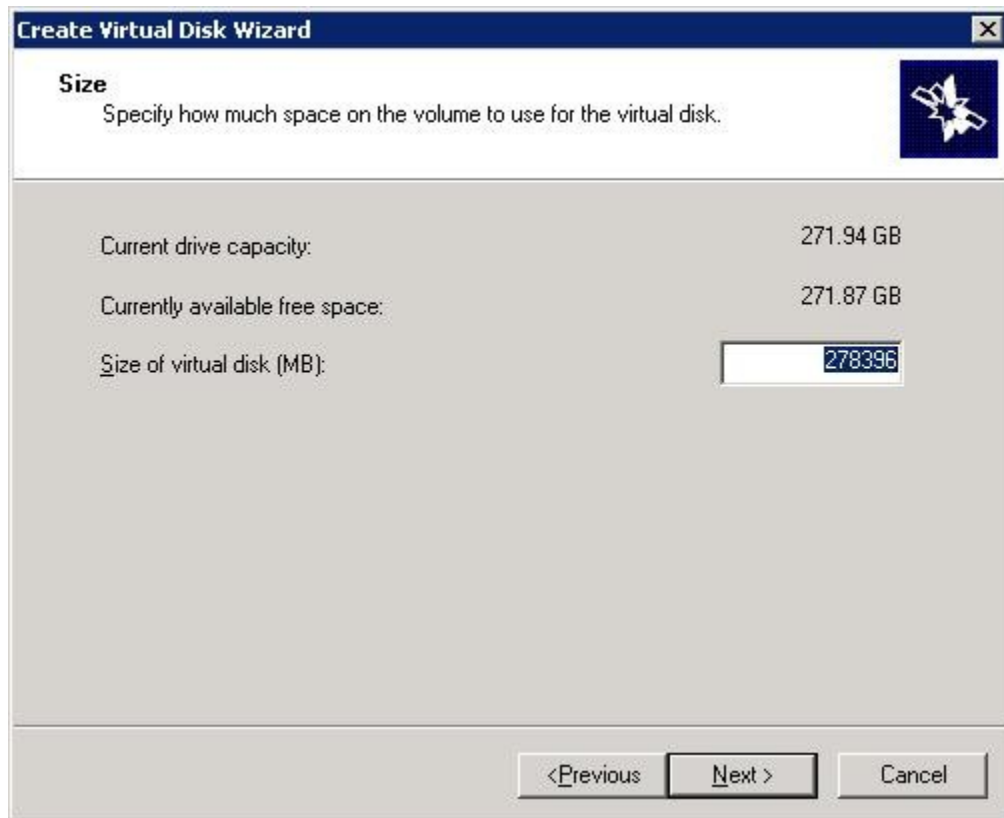
Figuur 15: iSCSI Virtual Disk maken

- We geven opnieuw een naam en locatie in:



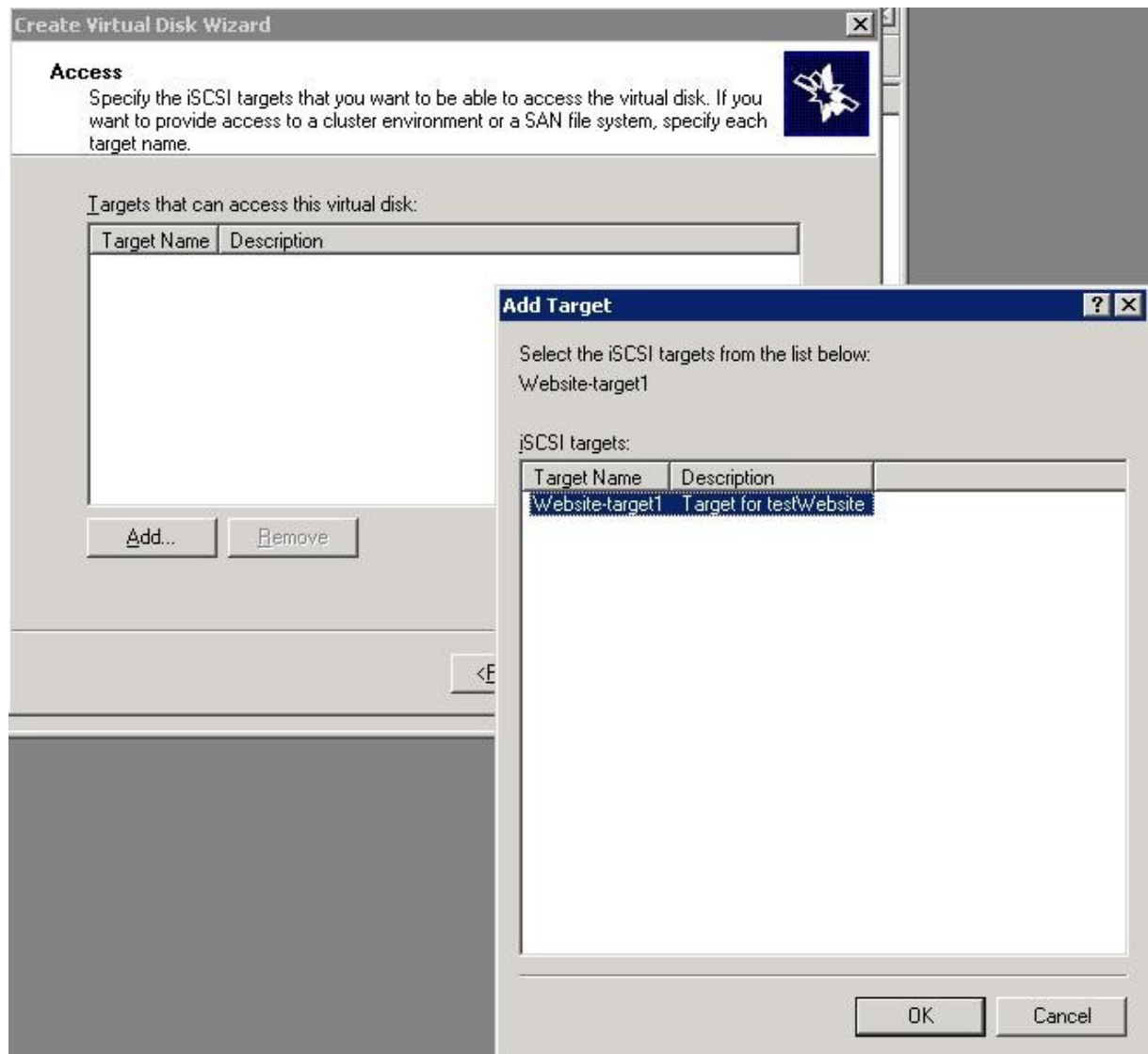
Figuur 16: iSCSI Virtual Disk wizard, naamgeving

- Daarna kunnen we de grootte kiezen. Hier kiezen we voor de maximale grootte, dit is natuurlijk niet verplicht.



Figuur 17: iSCSI Virtual Disk wizard, grootte

- Bij het volgende venster is er de mogelijkheid om een of meerdere targets te verbinden aan de schijf. Het is ook mogelijk om meerdere schijven aan eenzelfde target te verbinden.

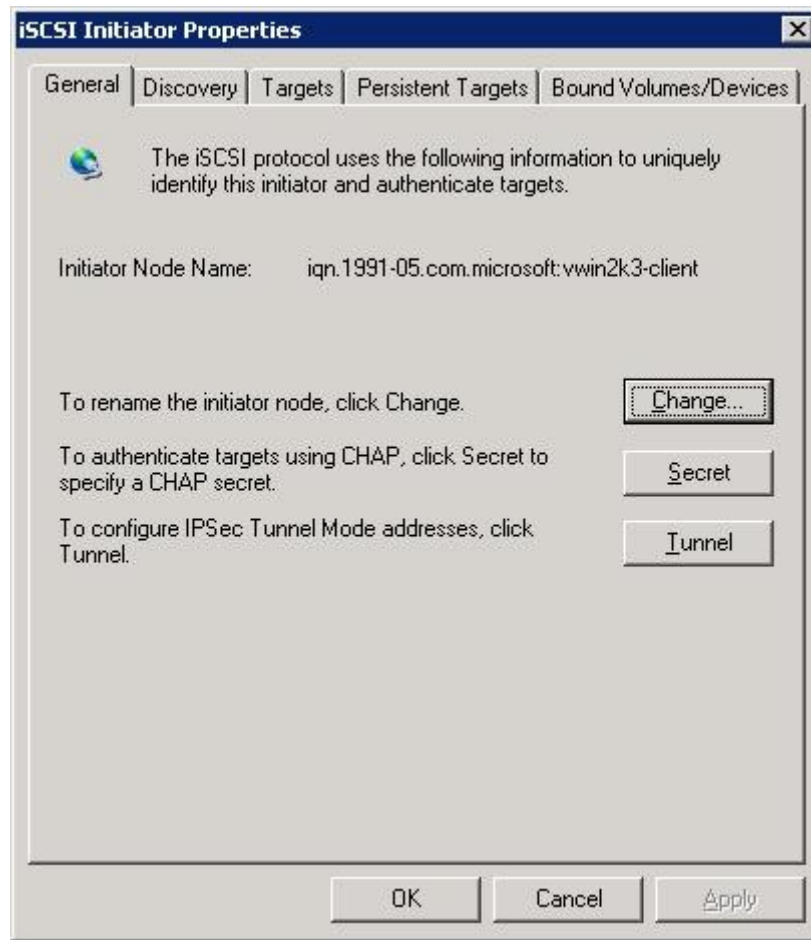


Figuur 18: Hier blijven enkel nog een druk op de “OK” & de “Finish” knop over.

2.4.2. Aanmelden met de Microsoft iSCSI Initiator

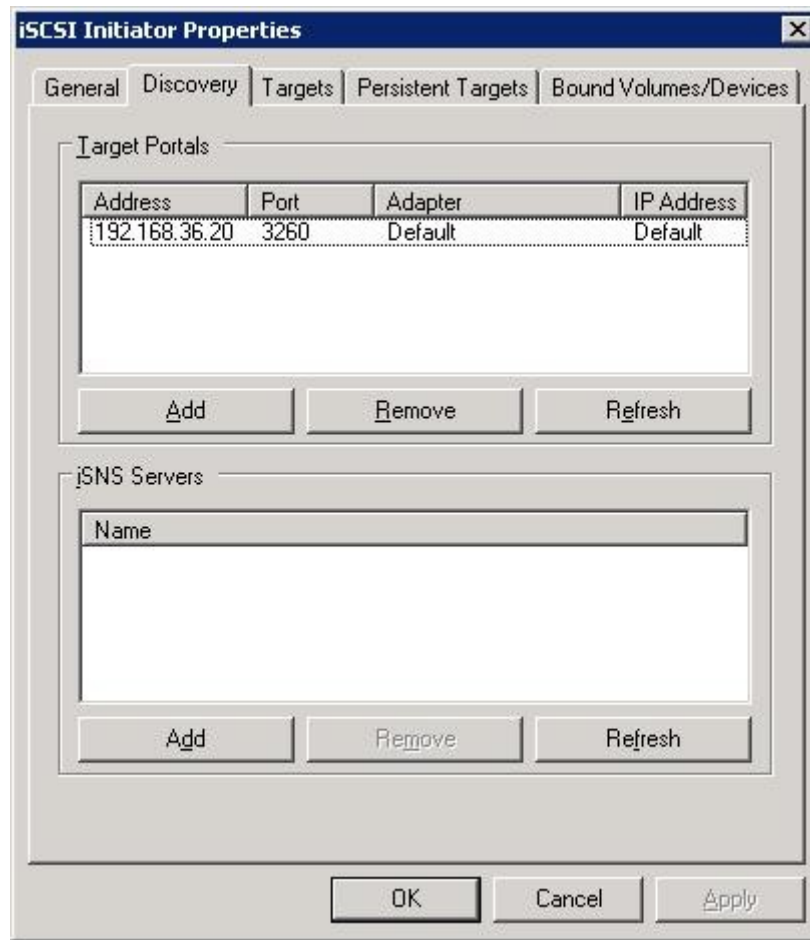
Alles is voltooid en de iSCSI Target is aangemaakt. Langs de initiator kant is het nu mogelijk om te verbinden, bijvoorbeeld met de Microsoft iSCSI Initiator. Dit is gratis software, af te halen van de Microsoft website.

- Bij het starten van deze software zien we meteen de IQN die hij heeft, we kunnen ook deze ingeven als goedgekeurde initiator bij de specifieke targets.



Figuur 19: iSCSI Initiator, IQN naam

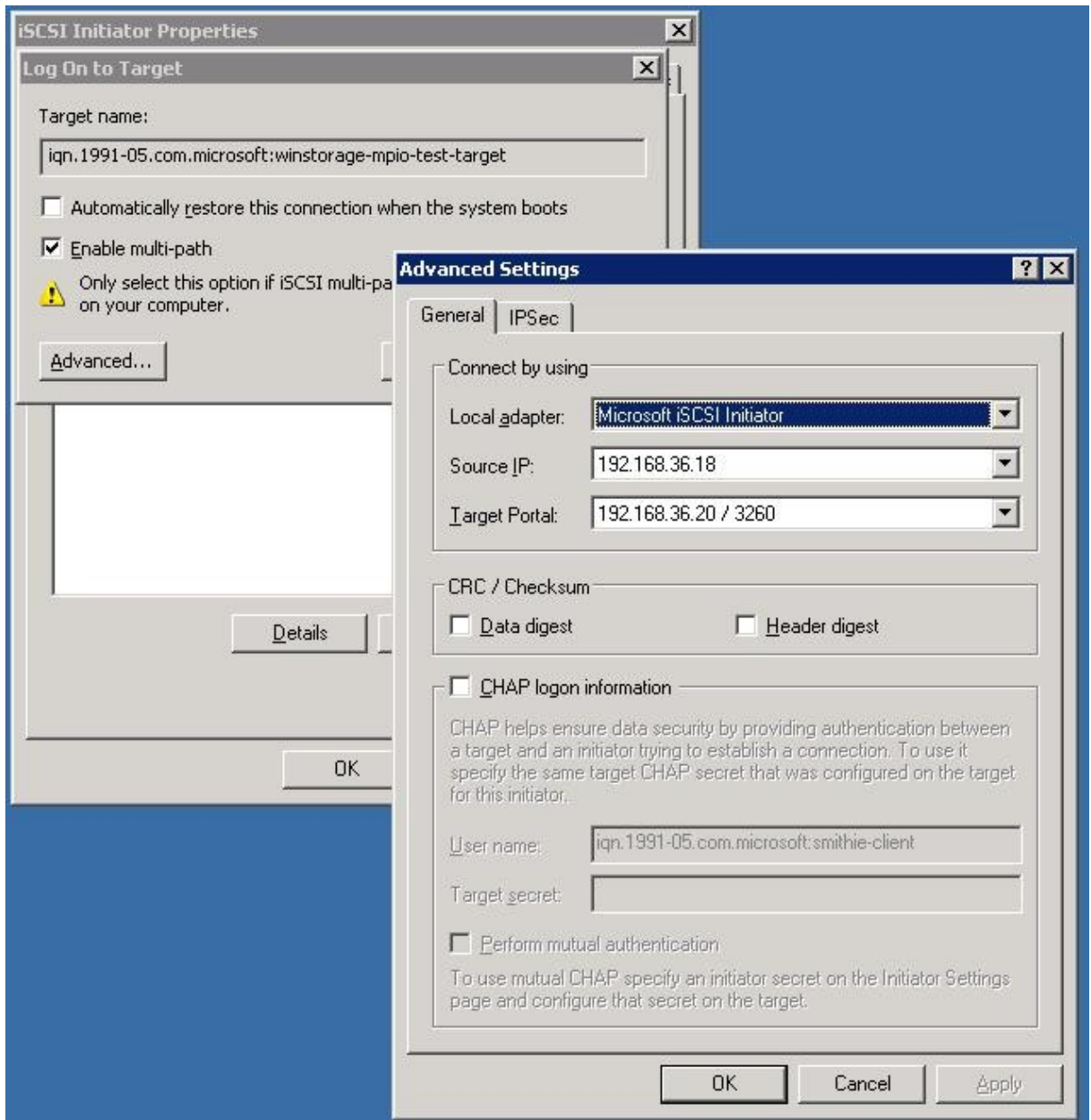
- Onder het tabblad "Discovery" dienen we de DNS-naam of het IP-adres in te geven van de server waar het iSCSI Target op draait. We klikken daarvoor op "Add".



Figuur 20: iSCSI Portal Discovery

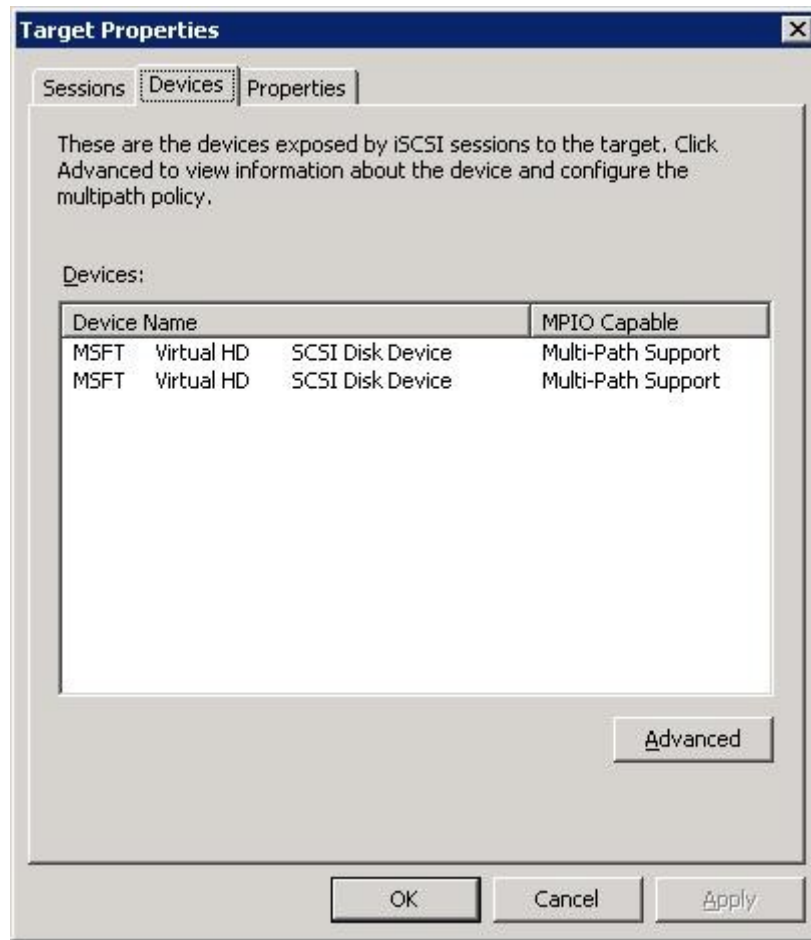
- Zoals zichtbaar bestaan er ook iSNS-servers, wat staat voor Internet Storage Name Servers. Dit is server-software (o.a. gratis te downloaden van de Microsoft-site), die voortdurend het hele netwerk afzoekt naar iSCSI servers. Op die manier is het niet nodig voor iedere nieuwe iSCSI Server een portaal toe te voegen aan de bovenstaande lijst.
- Na een druk op de knop "OK" kunnen we onder het tabblad "Targets" alle beschikbare targets zien, al dan niet na een druk op de knop "Refresh".
We kiezen voor Multi-Path IO, omdat dit de iets geavanceerde versie is om te verbinden. Zoals geweten is dit de mogelijkheid om meerdere netwerkverbindingen te maken met hetzelfde target. Dus als de client 2 (gigabit-)verbindingen heeft en de server heeft er ook 2, dan is in principe een dubbele doorvoersnelheid mogelijk. Later wordt dit getest, maar hier wordt het dus opgezet.
Concreet wil dit zeggen dat we meerdere keren inloggen en we kiezen bij iedere login welke netwerkkaart (langs de kant van target en/of initiator) we willen gebruiken. Voor het doelportaal dienen we hiervoor niet de 2 ip's mee te geven, 1 is genoeg. Bij het tabblad "Doelen" klikken we op het doel dat we willen en dan op "Aanmelden...".
- We krijgen een extra kader waarop we "Enable multi-path" aanvinken en dan klikken op "Advanced..." bij deze instellingen kunnen we dan bij "Local adapter" kiezen voor "Microsoft iSCSI-initiator". Dan kunnen we een source- en target-IP instellen. We

eindigen met een 2-tal keren op “OK” te drukken. Waarna we weer op het vorige scherm terechtkomen.



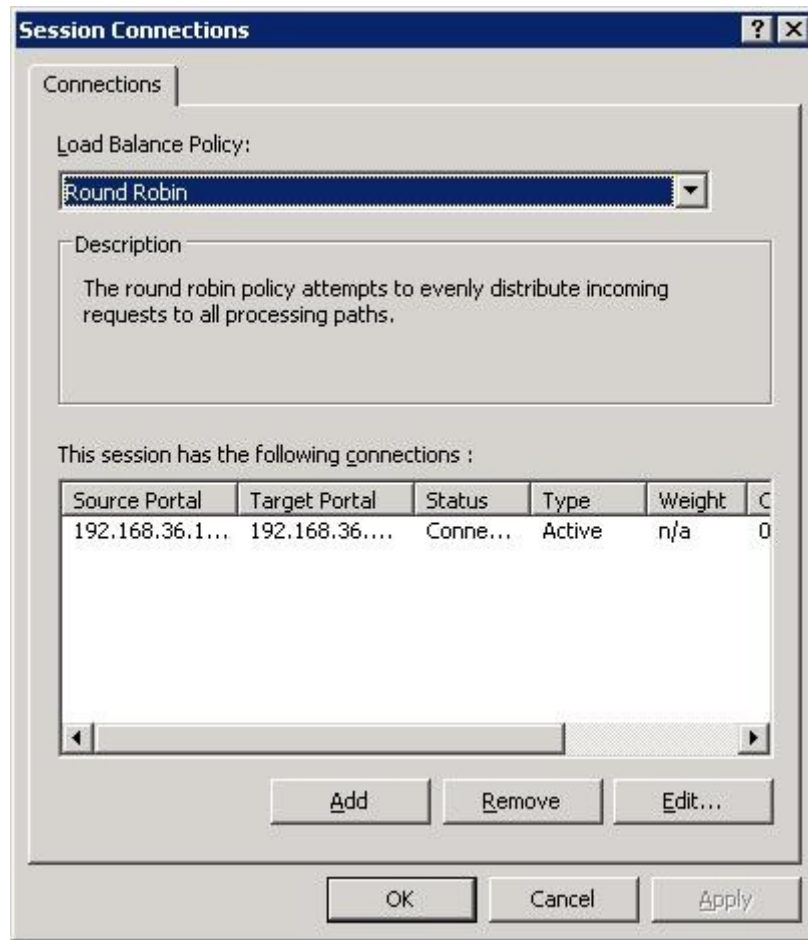
Figuur 21: iSCSI Initiator, MPIO aanmaken

- Daarna doen we alles nog eens over (selecteren, inloggen, multi-path aanleggen en een ander path kiezen).
- Als dit gedaan is, kiezen we nogmaals het target, maar nu kijken we naar “Details”. Als we daar naar het tabblad “Devices” kijken moeten we 2 devices zien met het detail: Multi-Path support.



Figuur 22: iSCSI Initiator, Multi-Path Support

- Tot slot gaan we het sessie-management-beheer nog instellen op “Round Robin”, dit doen we door in het Details-venster te kiezen voor het tabblad “Devices” en dan op “Advanced” te klikken. Terug tabblad “MPIO” en daar zien we de 2 (of meer) verbindingen. Daar kunnen we het beheer instellen.

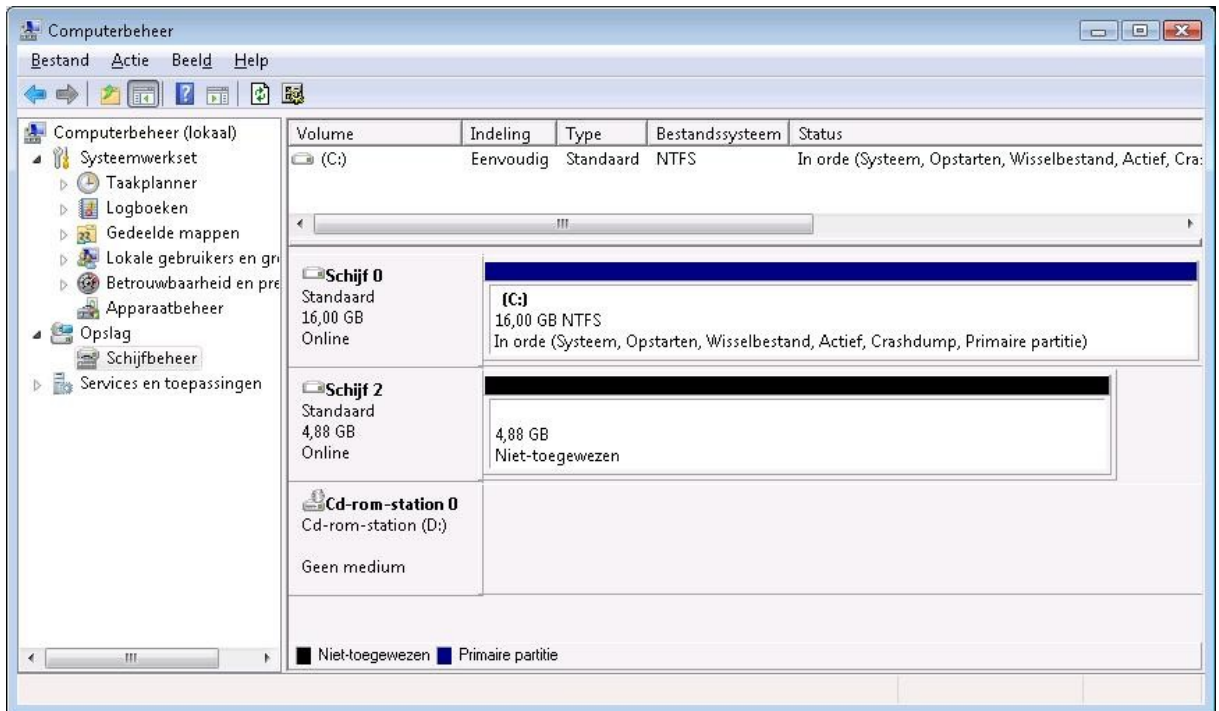


Figuur 23: iSCSI Initiator, MPIO opties

- Mogelijke optie's zijn Failover, Round Robin, Round Robin met Subset, Minste Wachtrijdiepte en Gewogen paden.
Failover wil zeggen dat er slechts 1 pad gebruikt wordt terwijl het 2^{de} in standby staat te wachten tot de eerste uitvalt, dan neemt de 2^{de} over.
Round Robin is een geavanceerde vorm van Failover, maar zal ten alle tijde de 2 paden gebruiken.
Round Robin with Subset is vooral handig bij meer dan 2 verbindingen, er kan dan aangegeven worden welke paden voor de Round Robin mag gebruikt worden. De andere zijn aangeduid als Failover.
Minste Wachtrijdiepte zal de meestgebruikte I/O-aanvragen over het pad sturen dat dit het best aankan (bijvoorbeeld grote bestanden versturen over een 1GBps pad terwijl een 2^{de} megabit pad gebruikt wordt voor kleine bestandjes).
Gewogen Paden, laat de gebruiker toe om gewichten toe te kennen aan verschillende paden. Dit zijn prioriteiten, een hoger cijfer geeft aan dat de prioriteit laag is.
Alles is nu opgezet om te kunnen testen, zoals bijvoorbeeld met de tool "SQLIO".

2.4.3. Benchmark opzetten (SQLIO)

- Nu zien we in schijfbeheer de betreffende schijf staan en kunnen we deze formatteren en partitioneren.

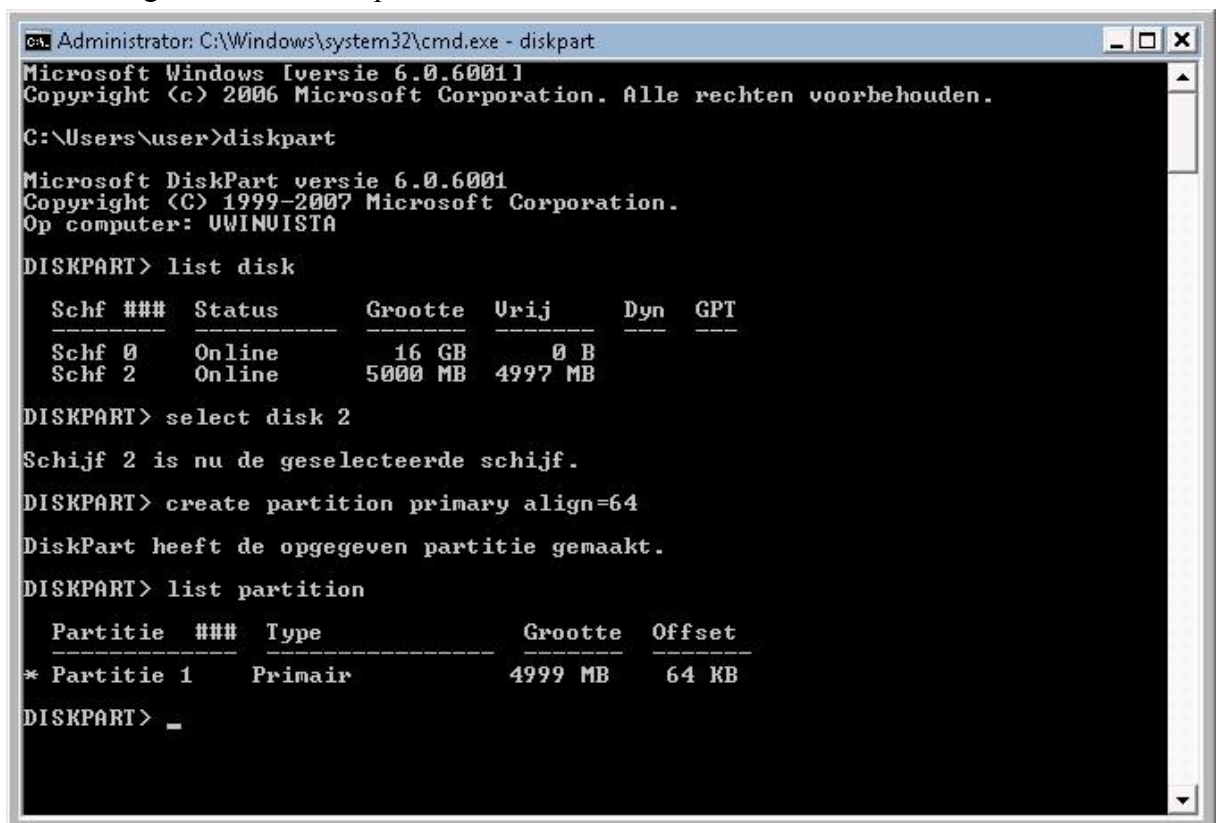


Figuur 24: SQLIO, schijfbeheer

- Dit partitioneren dienen we met de tool “diskpart” te doen, deze zit vanaf windows xp standaard ingebouwd en is bereikbaar via de command prompt. De reden voor het gebruik van deze command line tool is dat we de partition offset (zie prestatiefactoren) kunnen meegeven. Dit kan als volgt:

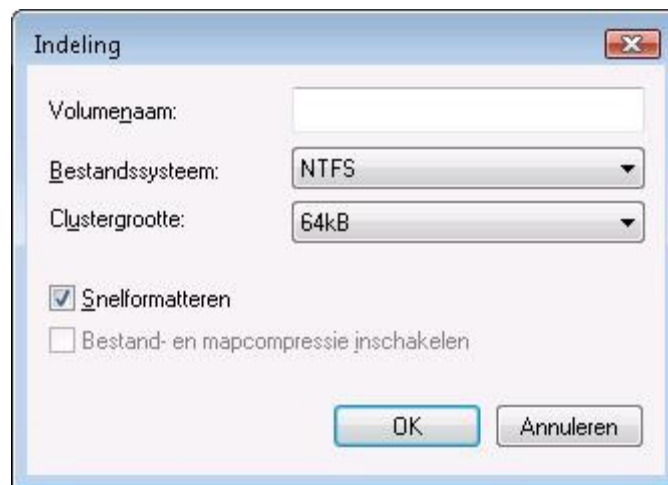
create partition primary align=64

Dit is de regel om de offset op 64KB te zetten.



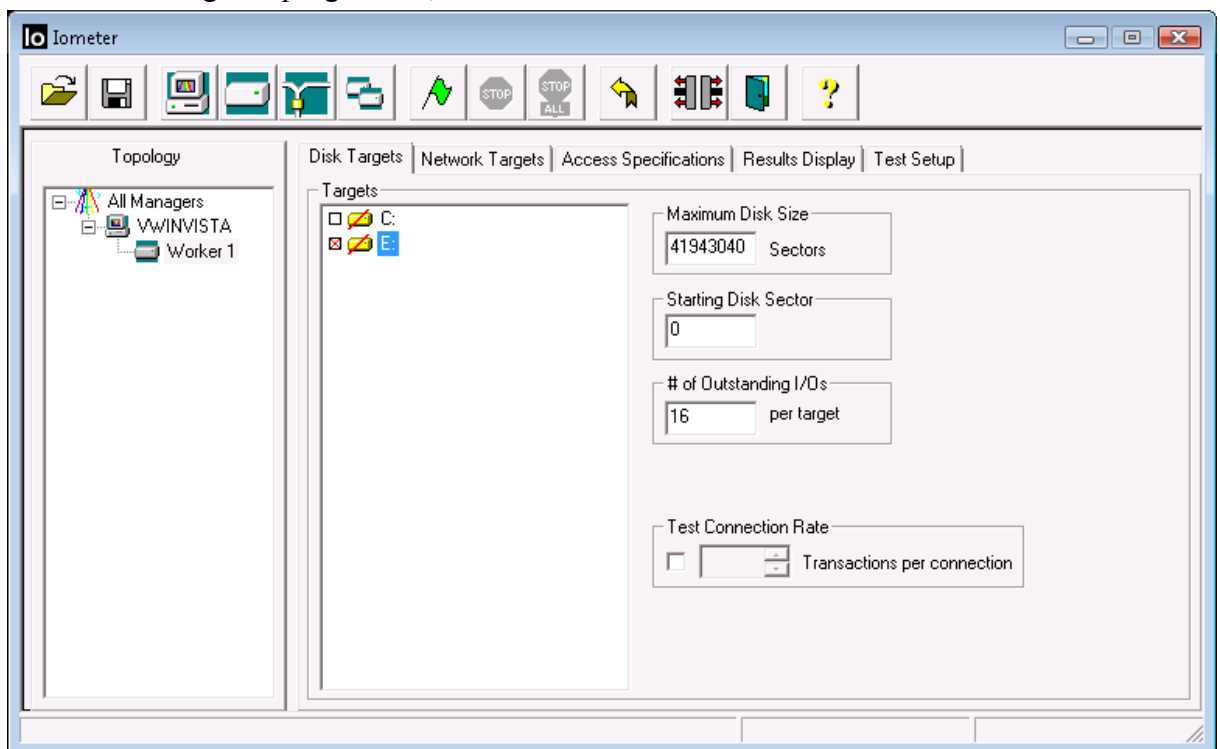
Figuur 25: SQLIO, Diskpart Tool

- Daarna volgt het formatteren en een schijf-letter toevoegen, we nemen 64KB als clustergrootte.



Figuur 26: iSCSI Initiator NTFS, Clustergrootte instellen

- SQLIO heeft een testbestand nodig van een bepaalde grootte, maar kan dit niet zelf aanmaken. IOMeter kan dit wel, hij maakt namelijk een testbestand 'jobw.tst' aan van een vaste grootte en vult dit op met nullen. Hierop kunnen we dan testen. IOMeter is een gratis programma, en we kiezen voor een standaard installatie.



Figuur 27: Dit is het hoofdvenster van IOMeter

Rechts gaan we het menu naar beneden tot we een worker kunnen selecteren, waarna we een schijf kunnen kiezen waarop we willen testen. Als er geen testbestand werd gevonden wordt er hij een rode streep door het icoon getrokken en zelf een bestand aangemaakt. Dit kan door de "Maximum Disk Size" in te stellen. In bovenstaande afbeelding staat ingesteld om een testbestand aan te maken van 20GB. We drukken op de groene vlag en zien dan een boodschap "Preparing Disks". Als dit voorbij is, is het

bestand aangemaakt en kunnen we IOMeter terug sluiten.



- SQLIO is een command-line tool. We hebben voor het gemak een batch-file gemaakt die als volgt is opgesteld:

We doen 6 sequentiële runs en 4 random, als parameter voor de test dient enkel een driveletter (bijv D:\) meegegeven te worden. Tussen iedere run zit er een regel 'timeout 30', om zeker te zijn dat ze elkaar niet beïnvloeden. 1 regel is als volgt opgebouwd:

```
"C:\Program Files\SQLIO\sqlio" -s60 -b8 -LS -o24 -fsequential %1\iobw.tst > result.txt
```

Eerst spreken we sqlio aan, en de -s staat voor de tijd (60 seconden), -b is de blockgrootte en deze laten we onder andere variëren. -LS duidt aan dat de uitkomst inclusief latencies moet getoond worden en -o24 zijn de outstanding IO's (hier voor een array van 12 schijven dus). -fsequential kan ook -frandom zijn. %1 is de driveletter gevolgd door de naam van het bestand. Daarna slaan we alles op in het bestand 'result.txt'. De volgende resultaten hebben een '>>' teken, zodat we het niet overschrijven.

- 1 typisch testresultaat die dan in result.txt zichtbaar zou moeten zijn is dit:

```
sqlio v1.5.SG
using system counter for latency timings, -1094757296 counts per second
1 thread reading for 60 secs from file F:\iobw.tst
    using 8KB sequential IOs
    enabling multiple I/Os per thread with 16 outstanding
using current size: 20480 MB for file: F:\iobw.tst
initialization done
CUMULATIVE DATA:
throughput metrics:
IOs/sec: 5277.06
MBs/sec: 41.22
latency metrics:
Min_Latency(ms): 0
Avg_Latency(ms): 2
Max_Latency(ms): 58
histogram:
ms: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24+
%: 13 63 2 2 3 3 1 1 1 1 1 1 2 1 1 1 1 1 0 0 0 0 0 0
```

2.5. Testen en resultaten

2.5.1. Opzet, testfactoren

Tijdens het project werden 2 belangrijke tools gebruikt, IOMeter & SQLIO. Deze programma's hebben (gelijkaardige) opties die moeten meegegeven worden voor de test start.

IOMETER

Deze heeft vele opties om verschillende schijven eventueel tegelijk te testen, met time-outs, verschillende groottes van testbestanden, lees-schrijf testen,... Bovendien houdt hij veel verschillende parameters bij, zoals het aantal I/O's per seconde, het gemiddelde, het maximum, het aantal fouten, CPU gebruik van de client ... Alles wordt ook opgeslaan in een csv bestand.

SQLIO

SQLIO is gemaakt door Microsoft en heeft als enig doel om data-requests van Sequel-Server te simuleren. Deze test gebruiken we om een real-time omgeving te simuleren. We nemen meestal dezelfde parameters over van IOMeter.

Testfactoren

Testbestand:

IOMeter kijkt of er een testbestand (naam='iobw.tst') bestaat op de schijf die geselecteerd is. Als dit niet het geval is zal hij een bestand aanmaken dat zo groot is als het aantal sectoren die opgegeven is, te beginnen op de sector die ook opgegeven is.

Wij laten deze start sector op 0 en de grootte van het testbestand staat voor de meeste tests op 41943040 sectoren. Als we weten dat 1 sector 512 bytes is, dan komen 41943040 sectoren overeen met ongeveer 20GB. We hebben vergelijkende tests gedaan waaruit bleek dat de grootte van het bestand voor de meeste tests geen groot verschil uitmaakt, met uitzondering van de cache-tests. Als we bij IOMeter de sectoren op 0 laten (standaard), neemt hij de volledige harde schijf in.

Aantal schijven & RAID-level:

Het aantal schijven waarmee we testen en het gebruikte RAID-level zijn uiteraard van kapitaal belang bij benchmarking. Als we verschillende technieken naast elkaar willen zetten moeten we daar rekening mee houden. 12 schijven in RAID0 (striping) zijn veel sneller dan 4 schijven in RAID-5.

Blocksize:

Is de grootte van de requests die uitgevoerd worden. Grotere pakketjes opvragen resulteert meestal in een hogere bandbreedte, in plaats van een heleboel kleine pakketjes.

Sequentiële & willekeurige toegang:

Sequentieel wil zeggen dat we allemaal blokjes na elkaar opvragen, wat uiteraard sneller werkt dan blokjes op willekeurige plaatsen van de schijf. Maar als we reëel willen testen, kiezen we best voor willekeurig.

Lezen & Schrijven:

Als we hier het reële beeld willen benaderen kiezen we best 33% schrijven & 66% lezen. Maar dit is niet mogelijk met SQLIO.

Outstanding IO's:

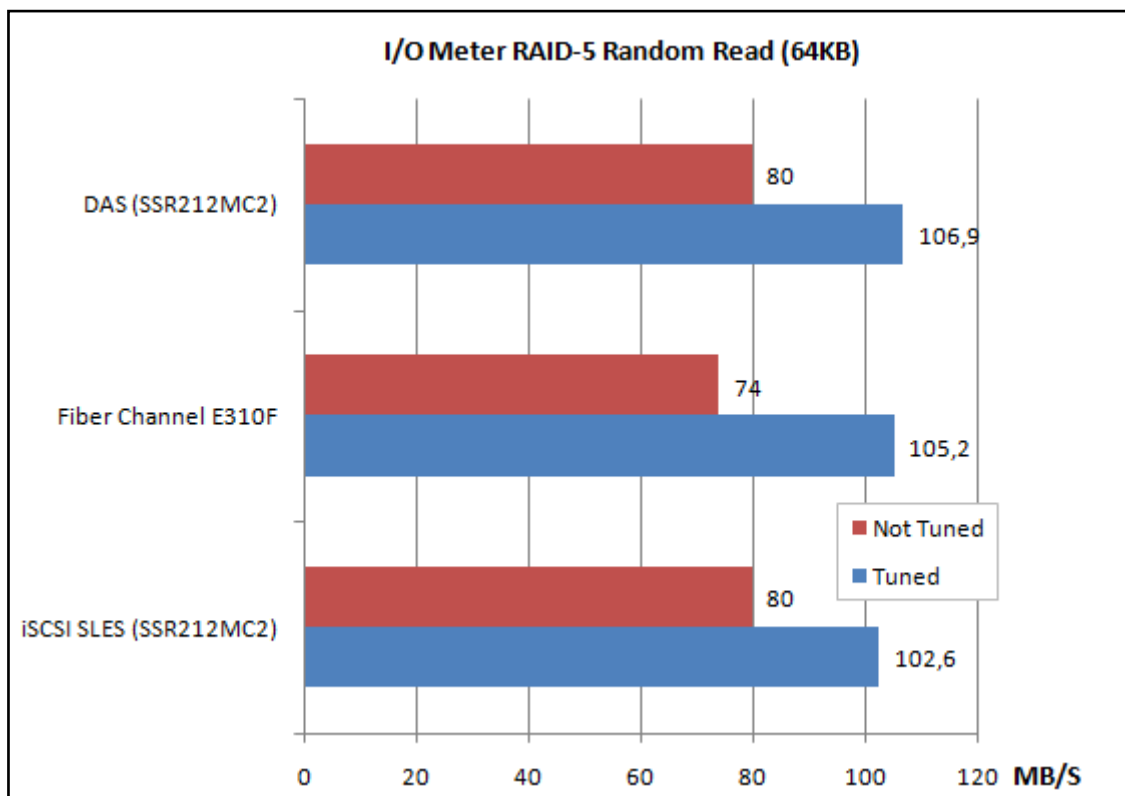
Dit is een aanduiding van het aantal requests die nog in de wachtrij staan bij de schijf (= Disk

Queue Length). Dit is het maximaal aantal dat SQLIO mag gebruiken. Indien we dit te hoog instellen dan hebben we latency problemen (= het te lang dueren eer er wordt geantwoord). Wordt dit te laag ingesteld, moet SQLIO te lang wachten voor hij de volgende request instelt en bijgevolg is de snelheid te laag. Een gunstige waarde is het aantal schijven maal 2.

2.5.2. Testresultaten

Uitleg: DAS is een rechtstreekse test binnen de machine, dus de RAID -configuratie via de RAID-controller rechtstreeks. Details van de gebruikte hardware is te vinden in Appendix D⁴.

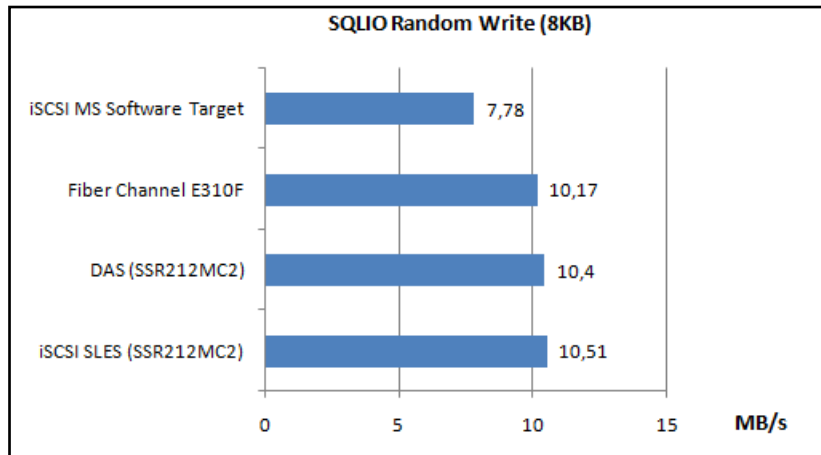
▲ Als eerste een overzicht van het voordeel als we testen met een correcte partitie offset.



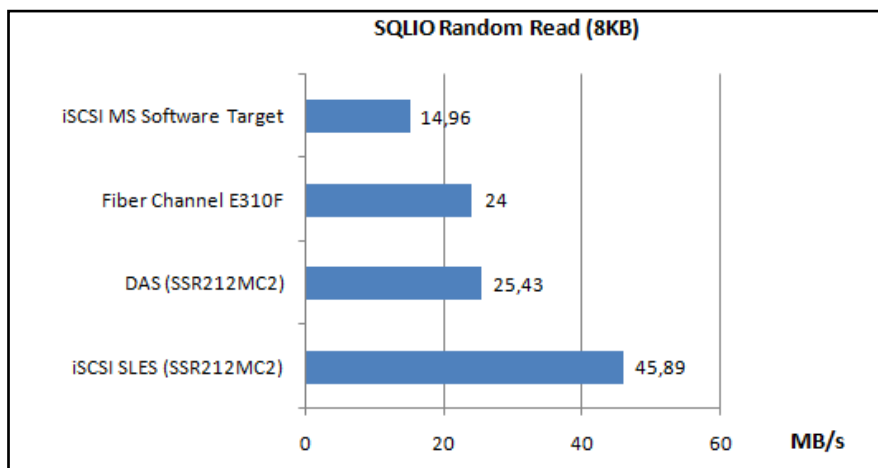
Figuur 28: Testresultaten; Partitie offset

⁴ Appendix D: Hardware Overzicht

- ⤴ Wat uiteraard ook zeer belangrijk is, is het verschil tussen Fibre Channel en iSCSI. We testen real-life, dit wil zeggen in RAID-5 (meest gebruikt), willekeurig lezen & schrijven in blokken van 8KB. Zoals zichtbaar wordt de beperkte bandbreedte van een gigabit verbinding (=125MB/s) toch niet gehaald, ook niet bij Fibre Channel.



Figuur 29: Testresultaten: FC tegenover iSCSI; Random Write

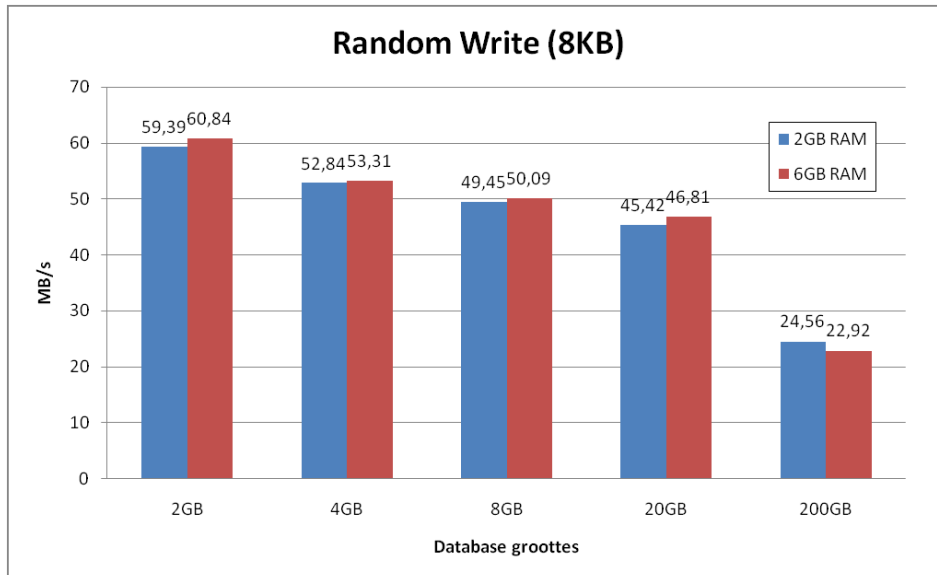


Figuur 30: Testresultaten: FC tegenover iSCSI; Random Read

Bij de laatste grafiek valt misschien op dat SLES zo goed presteert, dit komt (zoals zometeen zal blijken) doordat Linux (althans de gekozen Enterprise iSCSI Target) de gelezen data cached in het lokaal werkgeheugen, zodat een groot deel van de requests niet naar de harde schijf moeten gaan om data op te halen.

- ▲ Volgende grafiek toont het nut van leescache in het werkgeheugen, we hebben getest met bestandsgroottes (in de praktijk kunnen dat bijvoorbeeld databases zijn) van 2 tot 200GB.

En ook het werkgeheugen hebben we eens uitgebreid van 2 tot 6GB.

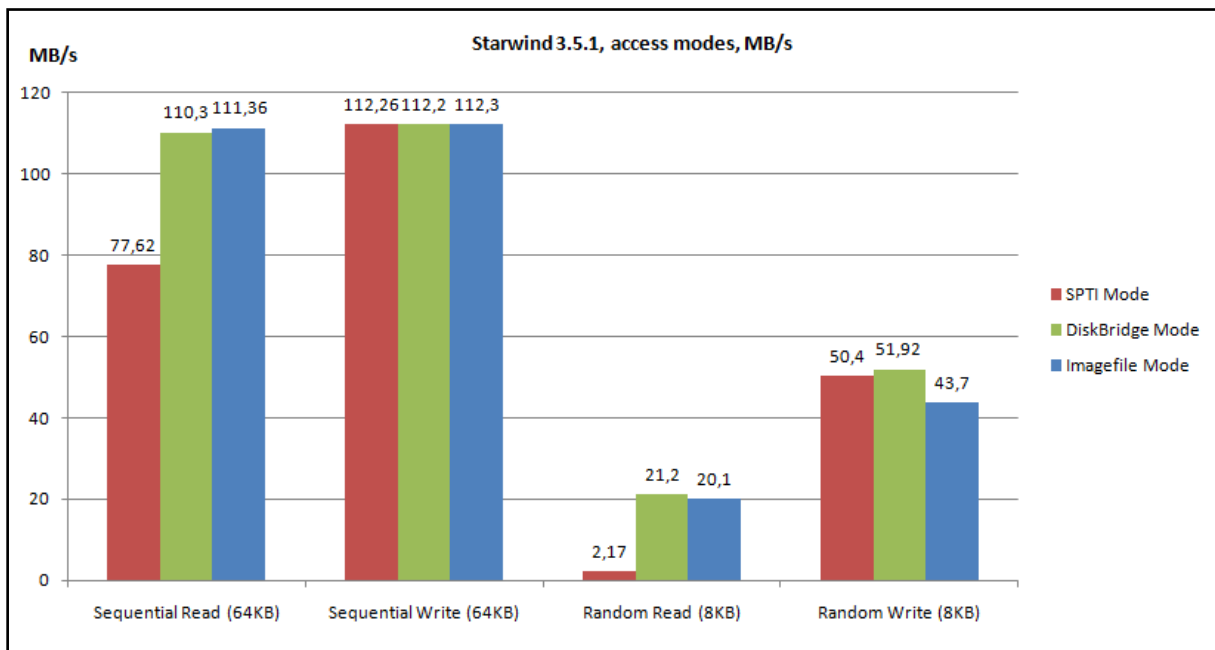


Figuur 31: Testresultaten: SLES Cache, Random Write



Figuur 32: Testresultaten: SLES Cache, Random Read

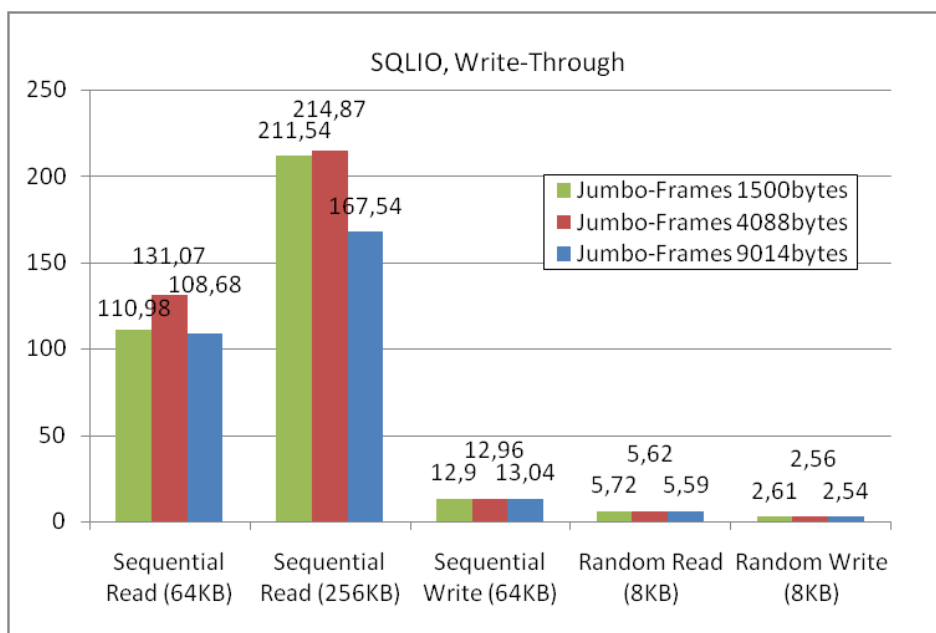
⤴ We hebben uiteraard ook de diverse access modes, die StarWind toelaat, getest.



Figuur 33: Testresultaten: Starwind Access modes

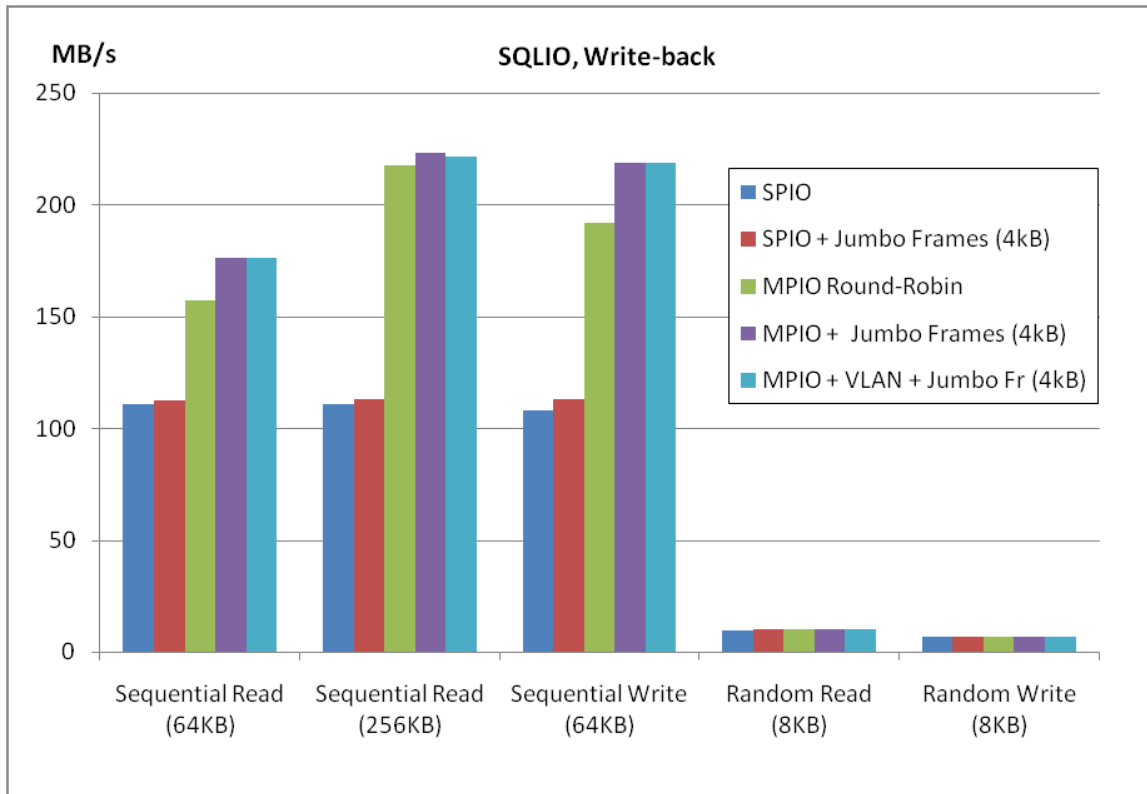
- De laatste tests binnen het Shared Storage gedeelte omhelzen het verder optimaliseren met behulp van netwerk-tools. De eerste resultaten-set toont ons welke Jumbo-frames we best instellen. Zoals bekend zijn er 3 mogelijkheden: default (1500 bytes), 4088 bytes of 9014 bytes. We zouden denken dat pakketten van 9KB het beste resultaat geven, maar dit valt een beetje tegen. Doordat we met SQLIO testen vragen we telkens datapakketjes van 8KB op, dit betekend dat als de tcp-pakketten 1,5KB zijn, we er 6 moeten versturen voor 1 SQLIO-datapakket. Als het 4KB is, moeten we er maar 2 versturen. Maar bij 9KB is er telkens 1KB die verloren gaat doordat deze niet gebruikt is. Hier treedt dus een klein verlies op, waardoor 4KB datapakketten het beste resultaat levert.

Alle volgende tests werden uitgevoerd met de Microsoft iSCSI Software Target op een RAID5-array van 4 schijven.



Figuur 34: Testresultaten: Jumbo-frames

- Tot slot een overzicht van de 2 overige optimalisaties, al dan niet gecombineerd met de eerste. Dit is een overzicht van jumbo-frames, MPIO & VLans. Zoals verwacht zijn er ook behoorlijke prestatiewinsten te behalen met MPIO, doordat de 2 (gigabit-)lijnen gebruikt worden. Het wordt pas echt leuk als we jumbo-frames combineren met MPIO, met gemeten snelheden tot 222,9MB/s. Echter het effect bij het creëren van een VLAN is zo goed als nihil, wellicht omdat ons netwerk niet zo actief is als bij sommige KMO's. Deze tests werden gedaan op een RAID5-array van 8 schijven.



Figuur 35: Testresultaten: Netwerk-tuning

3. Hardware Assisted Virtualization, de toekomst vandaag

Hier wordt dieper ingegaan op de diversie virtualisatie-oplossingen momenteel en in de nabije toekomst beschikbaar op de markt. Het doel is een inzicht verwerven in diverse oplossingen die server virtualisatie met zich meebrengt. Daarnaast worden 3 producten besproken die elk hun eigen visie hebben op server virtualisatie. Het doel is de 3 producten naast elkaar te zetten en te vergelijken. Dit wordt aangepakt door op 3 verschillende hardwareplatformen enkele real life situaties te schetsen en te testen. Ook hier weer met oog voor optimalisaties.

3.1. De theorie: wat is het en hoe werkt het?

Het bijzonderste aan virtualisatie is het feit dat het besturingssysteem (OS) niet weet dat het virtueel draait. Op dezelfde manier weet ook de hardware niet dat er virtualisatie is. Er is daar een hypervisor voor nodig. Hij nestelt zich tussen het OS en de hardware in en zorgt ervoor dat het virtuele OS niks doet met de hardware dat de andere besturingssystemen stoort. Bijvoorbeeld als een virtueel OS exclusieve toegang tot een randapparaat (bijv. cd-speler) nodig heeft, zorgt de hypervisor dat de toegang verdeeld wordt over alle besturingssystemen die de cd-speler nodig hebben.

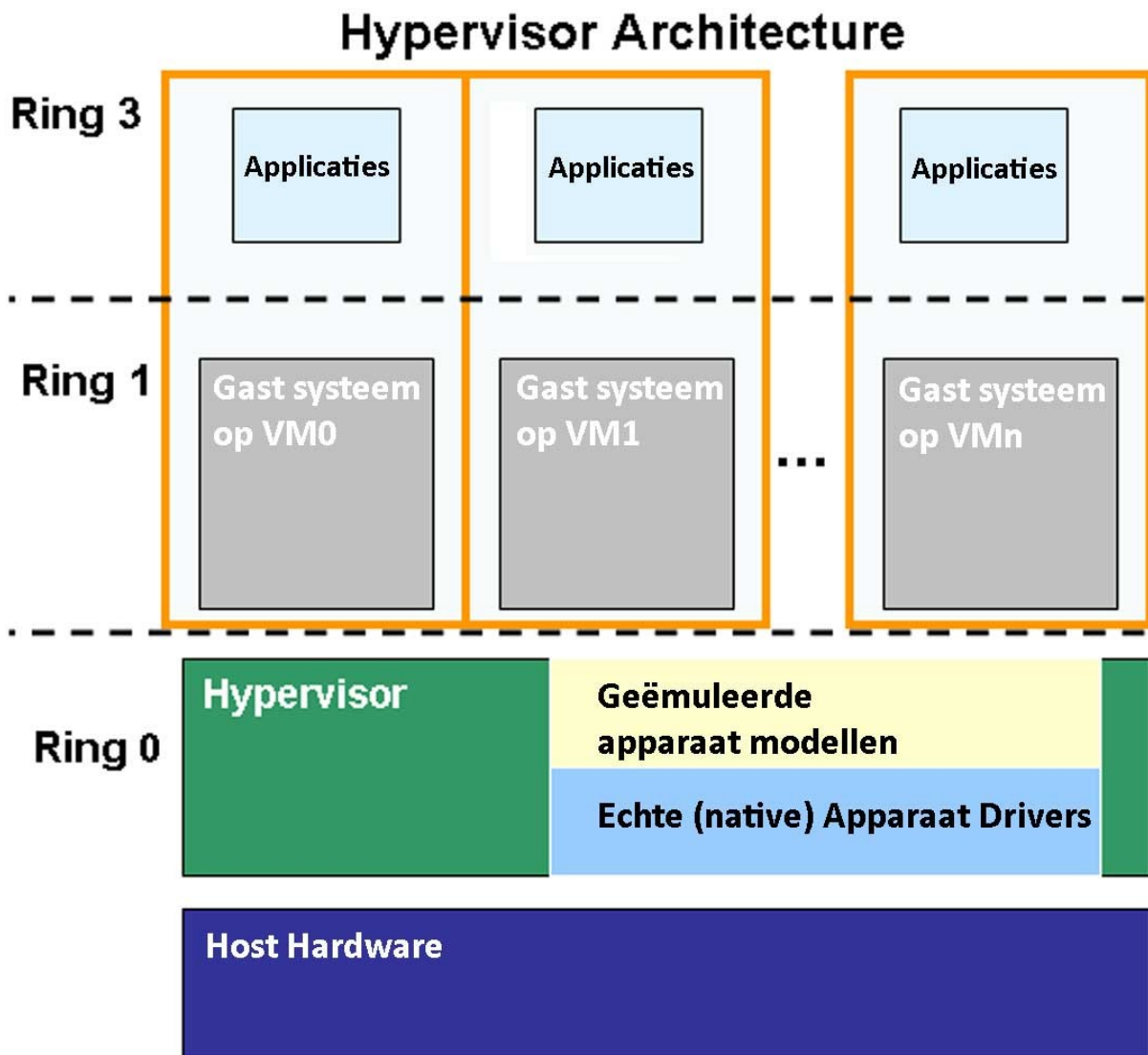
Wat is het grote probleem nu bij virtuele machines?

Alle besturingssystemen draaien in 2 modi, de kernel en de user modus. De kernel modus laat elke CPU instructie toe, inclusief de instructies zoals interrupts, geheugenbeheer. Dit is uiteraard de modus waar het besturingssysteem standaard in draait.

De user modus is minder privileged en laat enkel instructies toe om data te processen en berekeningen te maken. De meeste applicaties draaien in deze modus en moeten een zogenaamde system-call doen om toegang tot de hardware te krijgen.

Deze modi worden ook opgedeeld in ringen, ring 0 heeft de meeste privileges en is de kernel mode. Terwijl ring 3, zonder privileges, voor de applicaties is.

De techniek die alle software-gebaseerde virtualisaties gebruiken is het verschuiven van het besturingssysteem naar ring 1 en de hypervisor nestelen op ring 0. Dit laat de Virtual Machine Monitor (VMM) toe om het virtueel OS te controleren en alle system calls te onderscheppen.



Figuur 36: Ring privileges met software virtualisatie, de gast besturingssystemen (Guest OS) draaien niet langer in kernel mode (Ring 0), maar met minder rechten in Ring 1.

Maar de x86 structuur heeft een probleem, een x86 CPU luistert per definitie enkel naar instructies komende van ring 0 en sommige van deze instructies zijn niet zichtbaar voor de VMM.

Bijvoorbeeld de POPF instructie, die interrupts aan- en uitschakelt om ondersteuning te kunnen bieden aan oudere 16bit applicaties. Als deze wordt uitgevoerd door een OS in ring 1, dan zal de CPU deze negeren. Met als resultaat dat als het OS een interrupt wil onderbreken, dit niet lukt en de VMM helemaal niet weet dat dit gebeurt. In totaal bestaan er 17 van dit soort niet-detecteerbare instructies. Er zijn meerdere manieren om dit probleem op te lossen:

Uiteindelijk zijn er 3 manieren van virtualiseren: Software (of *full*) virtualisatie, Para-virtualisatie & hardware virtualisatie.

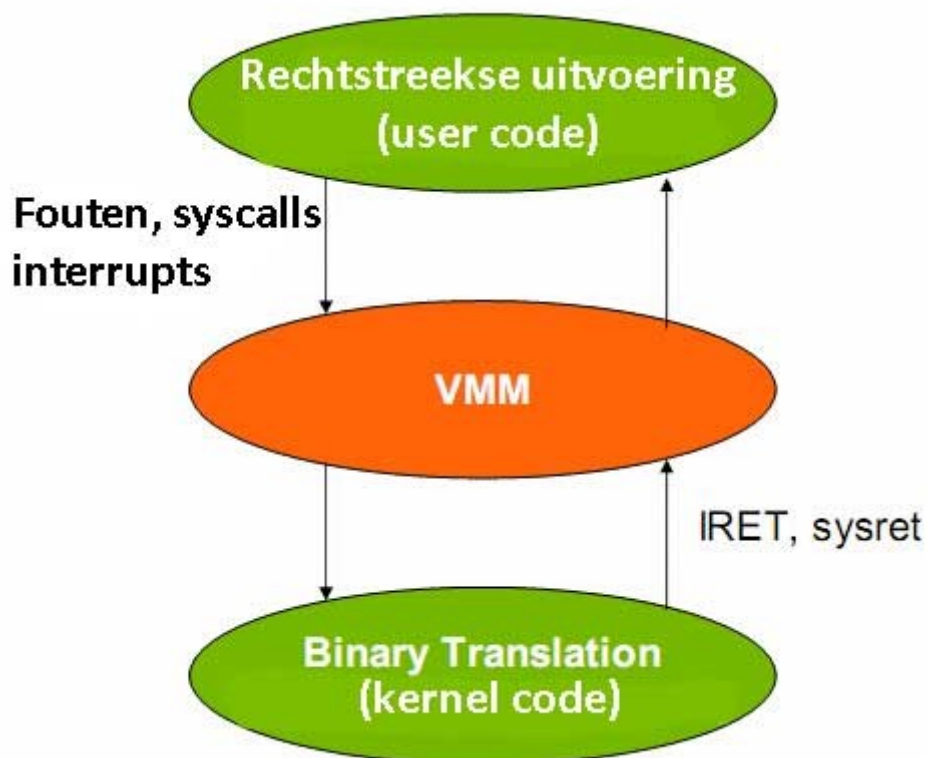
Bij full virtualization zorgt de hypervisor voor de volledige omzetting, zowel OS als de hardware weten niks van een virtualisatie en de hypervisor moet alles opvangen. Dit is de primaire manier waar VMware (ESX, Server, Workstation) standaard mee werkt.

Para-virtualisatie, door het aanpassen van de drivers en de kernel weet het besturingssysteem hier wel dat het virtueel draait. Het OS houdt hier rekening mee bij het geven van instructies aan de randapparaten. Dit is de voornaamste manier van werken voor Xen.

Hardware virtualisatie duidt dan weer aan dat de CPU (en eventueel chipset) zelf weet dat er virtualisatie wordt toegepast. AMD heeft daarvoor zijn AMD-V technologie, Intels variant heet VT-x.

3.2. Binary Translation (Software Virtualisatie)

VMware startte in 1999 met een eigen oplossing voor het probleem van de x86 verborgen instructies. Om die 17 instructies zichtbaar te maken hebben ze een techniek ontwikkeld die Binary Translation (BT) heet. BT vertaalt simultaan de code die de kernel van een guest OS wil uitvoeren en slaat deze op in een vertaal geheugen (Translator Cache). De code die echter van de applicaties komt, zal niet worden aangepast omdat BT er automatisch van uitgaat dat de user code veilig is.



Figuur 37: User code (ring 3) wordt rechtstreeks uitgevoerd. Binary Translation gebeurt enkel bij kernel code.

Enkel de kernel code moet door de vertaling. Anders gezegd kunnen we stellen dat de kernel van het OS niet langer zelf draait, het werkt slechts als een direct input voor de BT die zelf in ring 0 draait.

In veel gevallen is de BT een exacte kopie van de kernel van het OS. Enkel in sommige gevallen moet er een vertaling gebeuren waardoor de x86 code iets langer wordt dan normaal. Als de kernel een beveiligde instructie moet uitvoeren, zal de BT deze wijzigen in een iets veiligere code. Als de kernel toegang wil tot fysieke hardware, dan past de BT die code aan aan de virtuele hardware.

Het spreekt voor zich dat code vervangen door veiligere code veel makkelijker en performanter is dan de CPU fouten achteraf te laten opvangen. Toch wil dit niet zeggen dat er geen prestatieverlies is bij deze manier van werken.

Er is echter een groot nadeel met BT. System calls worden uitgevoerd door de applicaties en naar ring 0 gestuurd. Daar verwachten de applicaties om het OS te vinden, maar vinden daar de VMM. Die VMM moet de system calls dan emuleren, vertalen en doorsturen naar ring 1 waar het guest OS zit. Dan moet de VMM de controle even overdragen aan de kernel van het guest OS. Dit heet een SYSENTER.

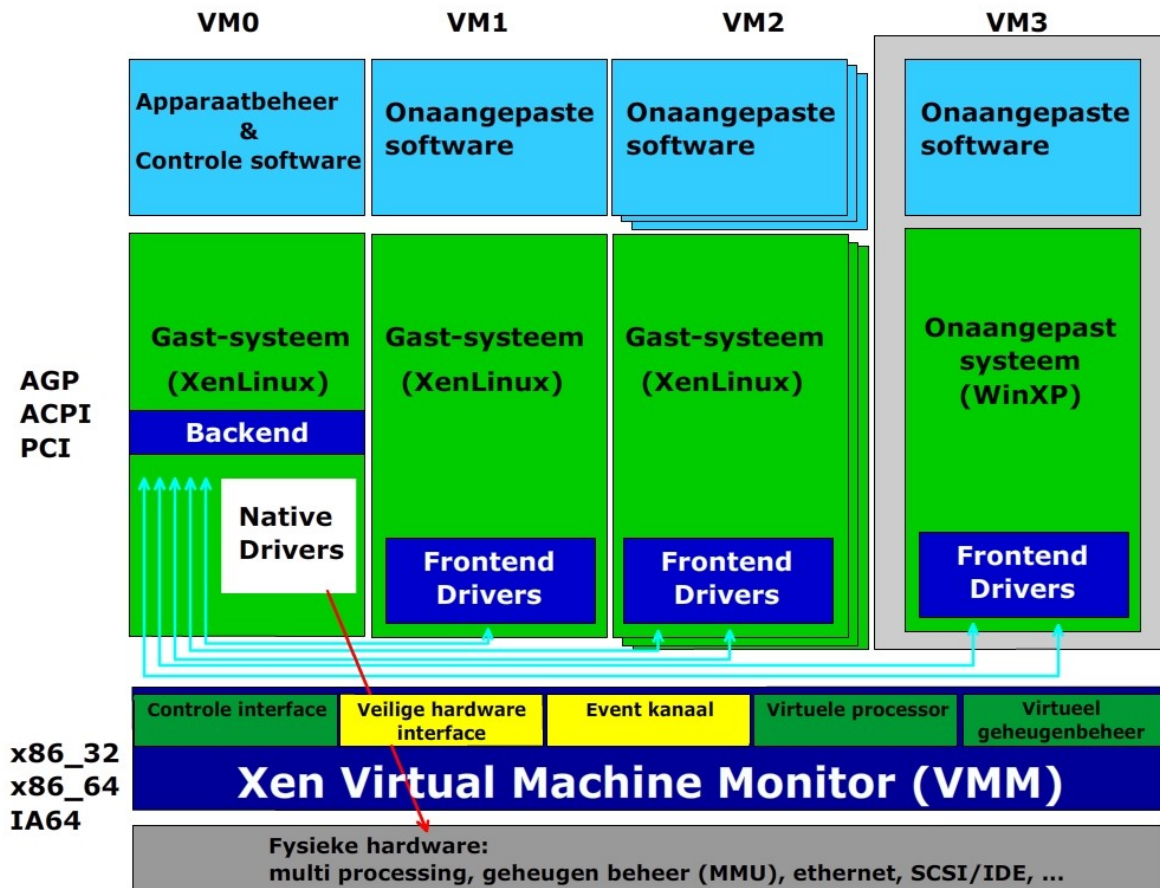
Als het guest OS zijn code uitgevoerd heeft, volgt er een SYSEXIT om terug te keren naar de applicatie. Maar het guest OS zit op ring 1 en heeft niet de nodige privileges om deze SYSEXIT uit te voeren. Dus wordt er een fout gegenereerd die de VMM moet opvangen en de resultaten correct emuleren naar de applicatie. Dit resulteert in een grote vertraging, VMware heeft dit opgemeten op een Pentium 4 van 3,8 GHz:

- ◆ Bij een native (niet gevirtualiseerd) systeem duurt een System Call 242 klok-cycli
- ◆ Met Binary Translation duurt dezelfde System Call (32-bit OS) 2308 klok-cycli

3.3. Para-virtualisatie

Veel verschil tussen Binary Translation en para-virtualisatie is er niet. Terwijl BT de gevaarlijke code simultaan verandert, zal para-virtualisatie dit doen in de broncode. Hierbij is er natuurlijk meer flexibiliteit mogelijk en er kan veel meer gevaarlijke code opgevangen worden.

De hypervisor zal enkel nodig zijn voor bepaalde kritische kernel operaties die het OS nodig heeft. Het meeste geheugenbeheer wordt door de guest OSes zelf gedaan. De hypervisor wordt enkel “aangeropen” voor dingen als page table updates en DMA toegang.



Figuur 38: Simpele drivers werken met de “normale” Linux drivers in VM0

De manier waarop bijv. XEN de geparavirtualiseerde drivers laat werken is als volgt; er is een beveiligde VM (genoemd Domain0) waar de native drivers inzitten. En de andere besturingssystemen die geparavirtualiseerd worden hebben gewoon een frontend interface die doorlinkt naar de native apparaat drivers. Dit betekent dat er geen emulatie is en dat de overhead dus minimaal is.

Dit maakt van paravirtualisatie een prachtconcept want het elimineert de volledige vertaalslag. Ook de I/O overhead wordt grotendeels aangepakt, en zelfs de system calls voelen een lichte verbetering. Er zijn echter grote nadelen:

- ◆ We kunnen geen onaangepaste besturingssystemen draaien (Windows is bijvoorbeeld nog onmogelijk bij paravirtualisatie).
- ◆ 64-bit besturingssystemen vormen een groot probleem, deze hebben extra kernel instructies en om het OS te beschermen kunnen we deze enkel in ring3 draaien.

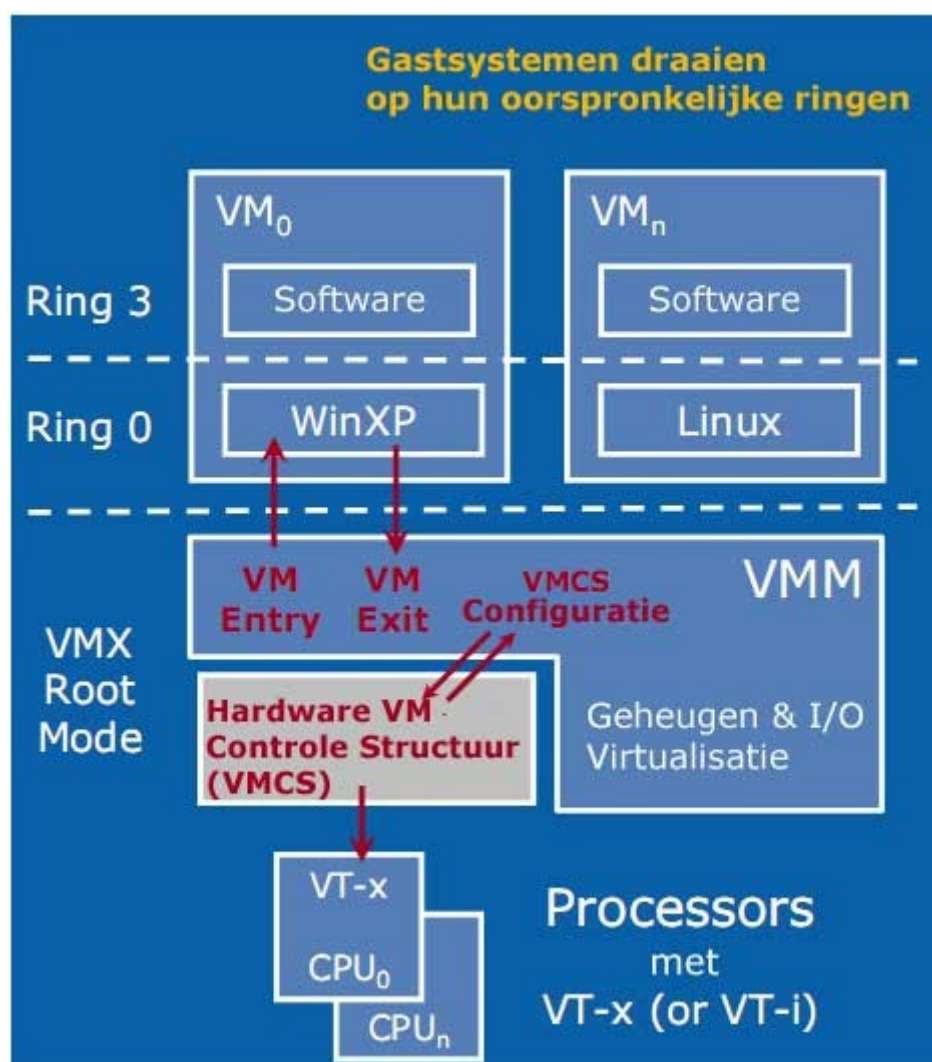
3.4. Hardware virtualisatie: Intel VT-x & AMD SVM

3.4.1. Eerste generatie

Hardware virtualisatie is niet zomaar een verbetering van Binary Translation of paravirtualisatie. De bedoeling is om alle fouten en beveiligde instructies op te vangen door een overgang te forceren van het gast OS naar de VMM. Dit heet een “VMexit”.

Uiteindelijk komt dit op het volgende neer: de CPU ontvangt de instructies van de besturingssystemen die normaal gezien de CPU of andere randapparatuur moet blokkeren. Maar in plaats van deze uit te voeren zal hij deze laten passeren langs de VMM (of hypervisor). Op die manier wordt het x86 probleem van de niet-detecteerbare instructies opgelost doordat de CPU deze instructies expliciet doorgeeft aan de VMM. Het voordeel is dat het gast OS terug op zijn bedoelde privilege mode draait, zijnde ring 0. Maar de VMM draait op een nog hoger privilege, namelijk ring -1 of “root mode”.

System calls zullen ook niet automatisch resulteren in een tussenkomst van de VMM. Zolang deze calls geen kritische code bevatten voor de CPU kan hij deze zelf laten afhandelen tussen de kernel en de applicaties. Wat uiteraard een heel groot voordeel is voor hardware virtualisatie.

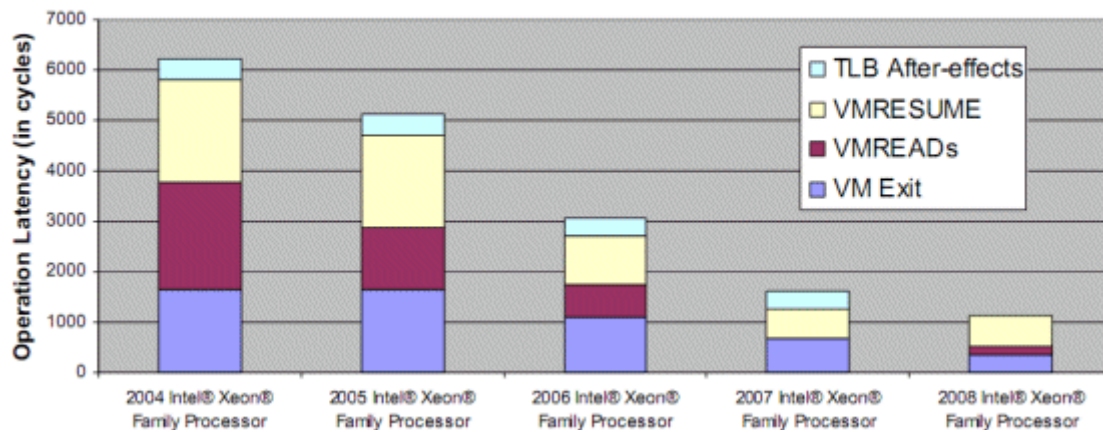


Figuur 39: Hardware Virtualization; het gast systeem staat terug waar het hoort: Ring 0.

Er is echter wel een probleem met Hardware Virtualisatie. Deze VMexit (gevolgd door een VMentry) kost behoorlijk wat vaste klokcycli. Het precieze aantal verschilt van architectuur tot architectuur en afhankelijk van het soort operatie (VMexit, VMentry, VMread ...) kan de kost oplopen van honderd tot enkele duizenden klokcycli.

Dit is uiteraard relatief, als het een system call betreft die sowieso al zeer veel klokcycli nodig heeft maakt de VMentry/VMexit niet veel verschil. Maar bij veel simpelere operaties en als dit veel gebeurt, kan deze overhead zeer snel oplopen. Bijvoorbeeld operaties als het creëren van processen, context switches, kleine updates voor de page tables nemen in een standaard (native) situatie maar enkele klokcycli in beslag. Maar bij hardware virtualisatie moeten deze via de CPU naar de VMM & terug en resulteren op die manier in een behoorlijk prestatieverlies. Binary Translation vertaalt deze operaties naar iets langere instructies maar met een kleiner prestatieverlies. Hetzelfde geldt voor paravirtualisatie, waardoor deze enorm veel sneller is dan hardware virtualisatie bij het afhandelen van deze kleine operaties.

Intel® VT-x Transition Latencies by CPU

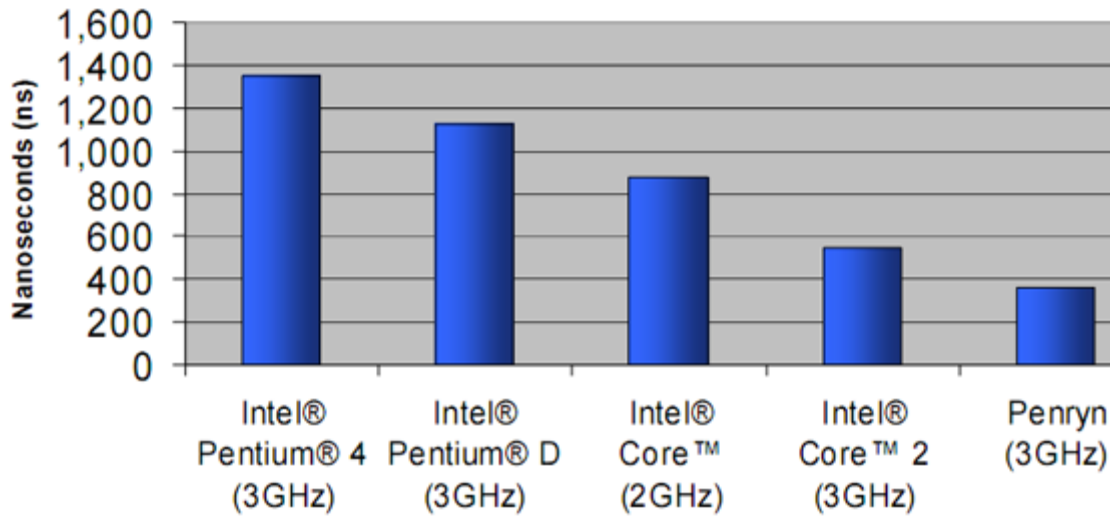


Figuur 40: VMentry & VMexit latency in de Xeon familie is verlaagd doorheen de jaren.

De uitdaging voor Intel & AMD is dus om dit aantal klokcycli zo laag mogelijk te maken. Doorheen de jaren is Intel daarin geslaagd met de Xeon familie.

Een tweede manier om de overhead te verminderen is om het aantal tussenkomsten van de VMM te verlagen. Dit kan door het aantal kritische instructies, die de CPU zelf kan afhandelen, te verhogen.

Round-trip (VMCALL + VMRESUME)



Figuur 41: VMentry & VMexit nummers (in nanoseconden) voor verschillende intel families.

We kunnen de vertraging die afkomstig is door hardware virtualisatie samenvatten in 1 grote formule:

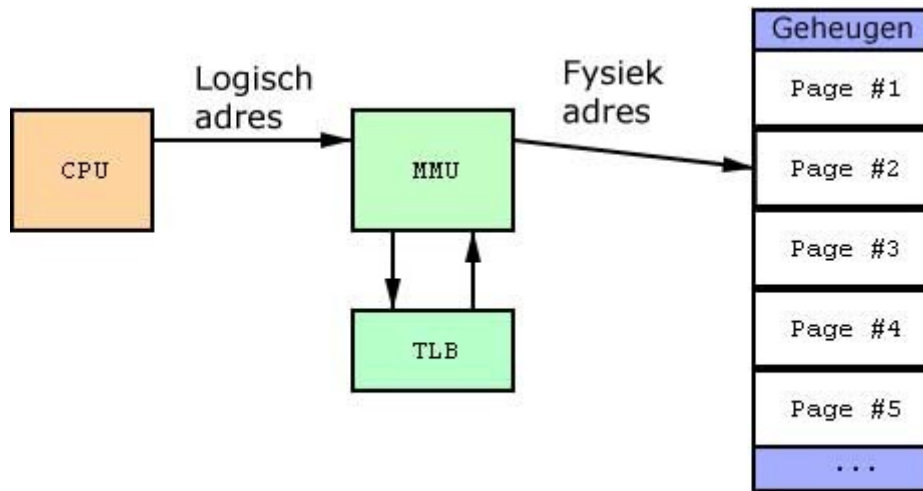
$$\text{Totale virtualisatie overhead} = \text{som van} \\ (\text{frequentie van overgangen van VM naar VMM} \\ \times \\ \text{vertraging van één zo'n overgang})$$

3.4.2. Tweede generatie: Nested of Extended Page Tables

Om het probleem van de page tables te begrijpen, bespreken we eerst het geheugenbeheer van een besturingssysteem.

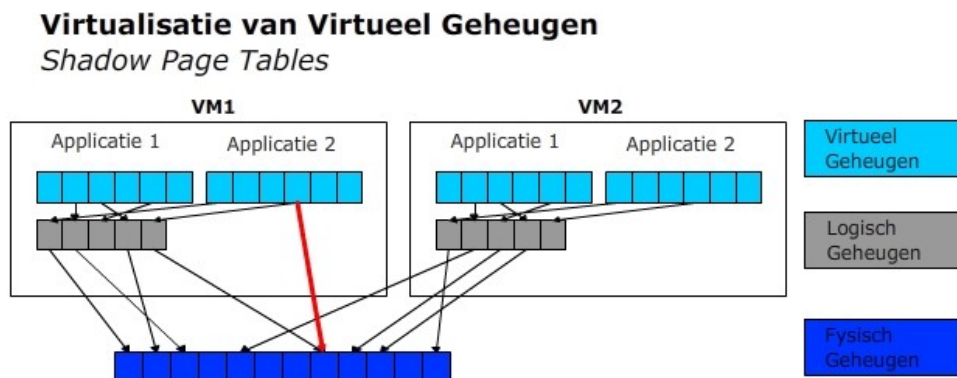
Het geheugen dat een applicatie toegewezen krijgt is nooit een effectief stuk geheugen, maar daarentegen wordt er gewerkt met logische adressen. De Memory Management Unit (MMU) zet deze dan om naar fysieke geheugenadressen. Een CPU heeft een adresruimte die wordt ingedeeld in zogenaamde pages, daarin worden de logische adressen gekoppeld aan de correcte fysieke adressen.

Om deze omzetting zeer vlot te laten verlopen beschikt de MMU over een zogenaamde Translation Look-aside Buffer (TLB). Dit is een soort cache in de CPU en de meestgebruikte table pages worden in deze buffer opgeslagen. Doordat ruimte op een CPU nogal beperkt is, kunnen het aantal pages in een TLB beperkt zijn tot 64 à 1024 items. Als een page niet gevonden is in de (supersnelle) TLB dan moet de MMU in zijn eigen (tragere) page-table zoeken.



Figuur 42: Schematische voorstelling van MMU, TLB, CPU & geheugen.

Bij een virtuele machine heeft het gast OS zelfs geen toegang tot deze logische adressen (en de TLB). Hij heeft daarentegen wel te maken met een geëmuleerde MMU in de VMM die werkt met zogenaamde Shadow Page Tables die nog steeds in de echte MMU zitten. De echte MMU doet nog steeds het werk, maar dan wel op de Shadow Page Tables. Deze manier van werken is echter zeer CPU intensief en kost tussen de 3 & 400 keer meer klokcycli dan een native manier van werken.

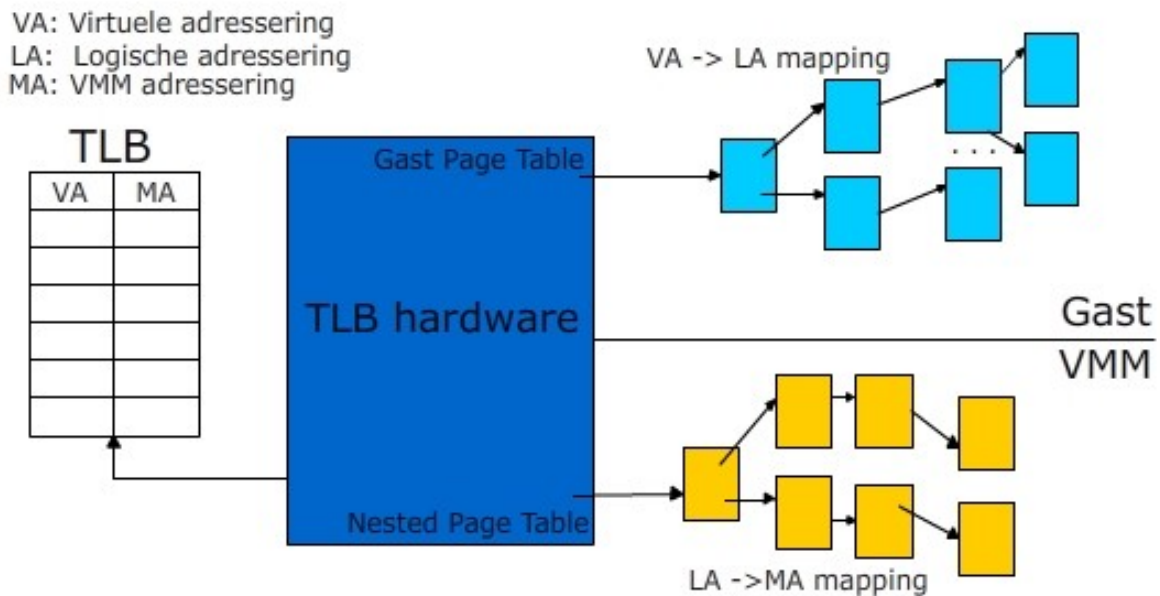


Figuur 43: Virtualisatie van Virtueel Geheugen, Shadow Page Tables

Bij voorgaande afbeelding is duidelijk dat de dikke, rode lijn veel sneller te berekenen is dan de zwarte lijnen. Dit is het voordeel van Shadow Page Tables. Er is echter een probleem; telkens als het OS een kleine update doet aan zijn page tables moet er een (kleine) aanpassing gebeuren aan de Shadow Page Tables. En dit is slecht nieuws voor de virtualisatie oplossingen maar voor hardware virtualisatie is er echter een oplossing op komst:

AMD's Nested (of Tagged) Page Tables & Intel's Extended Page Tables

Hun oplossing is om één hele grote TLB te maken die zelf bijhoudt bij welk virtueel OS een bepaalde TLB hoort. Deze doet dit aan de hand van een extra kolom in de tabel (vandaar Tagged Page Tables), deze nieuwe tag heet de *Address Space Identifier* (ASID).



Figuur 44: Nested/Extended Page Tables

Zoals getoond in figuur 44 houdt een CPU met hardware ondersteuning voor Nested of Extended Page Tables zowel de virtuele naar logische adressering bij (zichtbaar voor Gast OS), als de logische naar de VMM adressering (zijnde het echte fysieke geheugen). Uiteraard moet de TLB hiervoor groot genoeg zijn. Volgens AMD is er een vooruitgang van 23% te merken bij deze manier van werken. Dit is echter vooral handig als we met meerdere CPUs per VM werken, want onderling moeten deze hun page tables vaak synchronizeren. Dit zorgt ook voor flink wat overhead als er gewerkt wordt met Shadow Page Tables.

Er zijn echter wel nadelen aan Nested/Extended Page Tables; het opzoeken van een fysiek adres is veel complexer geworden. Telkens er iets moet opgezocht worden in de TLB hardware, moet er gezocht worden in zowel de gast mapping als in de VMM mapping.

Tweede nadeel is de standaardisatie; de twee technologieën onder Intel & AMD zijn niet onderling compatibel. Dit betekent dat software-ontwikkelaars aparte modules moeten ontwikkelen voor AMD en voor Intel. Dit resulteert in extra code in de software-pakketten die eigenlijk niet echt nodig is.

Een klein overzicht van welke CPUs nu welke ondersteuning bieden; HVT staat voor Hardware Virtualization Technology (eerste generatie dus), en EPT/NPT voor de tweede generatie.

Overzicht van CPUs met virtualisatie ondersteuning

Tabel 1: Tabel met CPUs die ondersteuning bieden voor hardware virtualisatie.

<u>CPUs met Virtualisatie ondersteuning</u>		
Processor	Type of Virtualization	Implementatie-graad (virtualisatie-snelheid)
Xeon 50xx, Xeon 70xx & Xeon 71xx	enkel HVT	Behoorlijk traag
Opteron Socket-F, Xeon 53xx	enkel HVT	Matig
Xeon 54xx	enkel HVT	Behoorlijk snel
Nehalem, Quad-core Opteron	HVT & EPT/NPT	Onbekend

4. VMware ESX, marktleider in virtualisatie

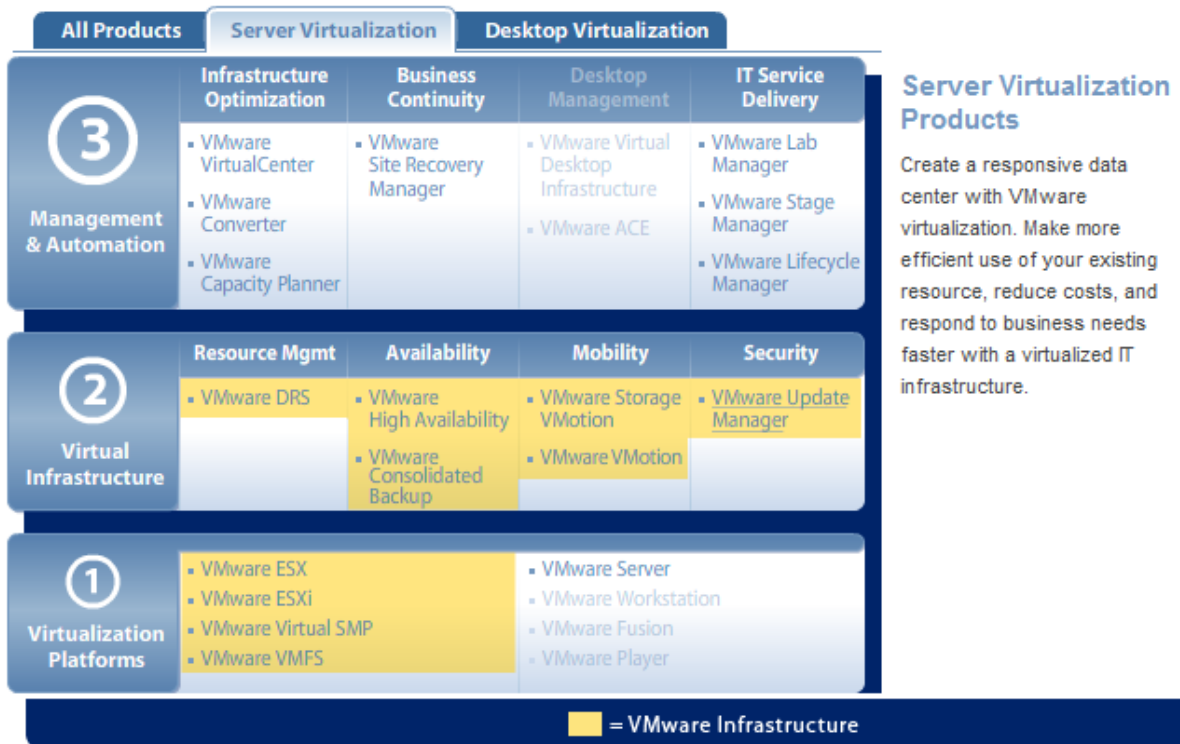
VMware werd opgericht in 1998 en is in zijn korte leven uitgegroeid tot de virtualisatiemarktleider. Met meer dan 100.000 klanten, 10.000 partners en een omzet van \$ 1,3 miljard is VMware één van de snelst groeiende software bedrijven ter wereld.

In 1999 kwam hun eerste product op de markt, namelijk VMware Workstation. Vooral bedoeld voor desktop virtualisatie werd het al snel populair voor developers, die op die manier hun software kunnen testen op virtuele machines zonder risico. In 2001 begon VMware aan de servermarkt en dit met de GSX en de ESX Server. Nog later kwam ook de gratis versie van de GSX, namelijk de VMware Server.

VMware houdt er een hele virtualisatie-filosofie op na. Deze heeft de naam VMware Infrastructure en bestaat uit vele onderdelen waaronder ESX Server en Virtual Center.

Binnen ons onderzoek gebruikten we vooral de Infrastructure 3i, want ESX is slechts in zeer uitzonderlijke gevallen apart te verkrijgen.

4.1. VMware Infrastructure 3i

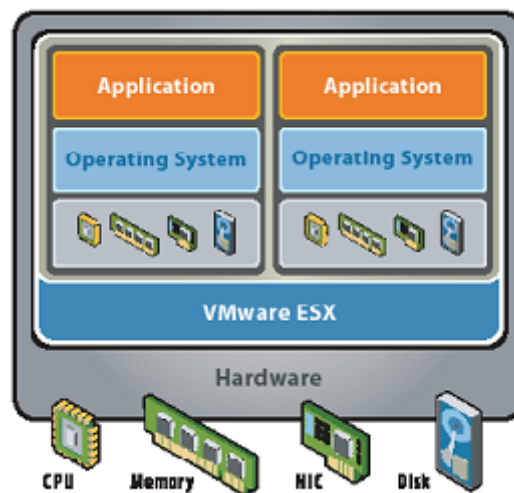


Figuur 45: Overzicht van VMware server producten: VMware Infrastructure

- ◆ VMware ESX is het hoofdproduct en is reeds in ontwikkeling sinds 2001. Het is eigenlijk de hypervisor die moet geïnstalleerd worden op de host (=de fysieke server). Op het moment van schrijven is de nieuwste versie ESX 3.5 en dat is tevens de versie waar we mee gewerkt en getest hebben. VMware ESX Server maakt gebruik van een gestripte versie van een kernel gebaseerd op onderzoek van Stanford University's

SimOS die na de hardware initialisatie het werk over neemt van de Linux kernel. De service console voor 2.x is afgeleid van een aangepaste versie van Red Hat Linux 7.2 en voor 3.x van Red Hat Enterprise Linux 3. Algemeen wordt deze service console gebruikt als een boot loader voor de vmkernel en voorziet in beheers interfaces (via Command Line, webpagina, of Remote Console).

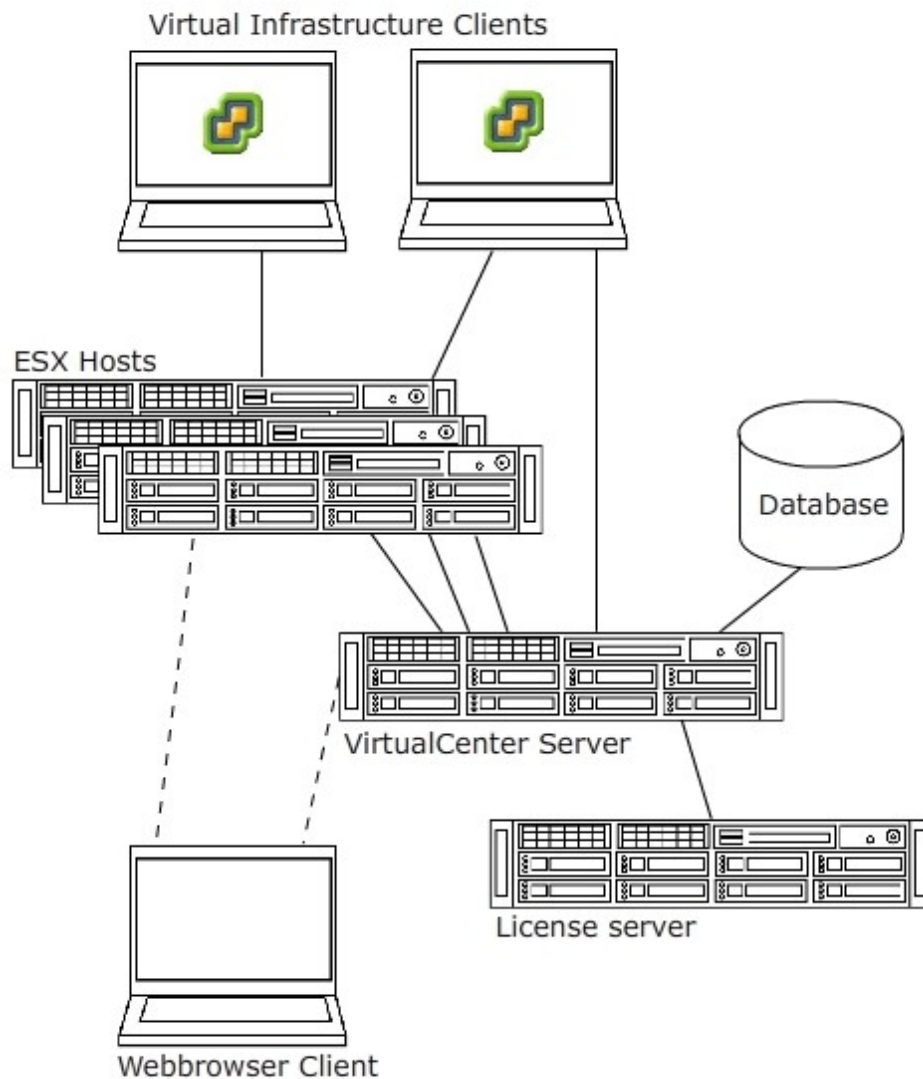
- ◆ VMware ESX is van de hypervisor familie (in tegenstelling tot Workstation of VMware Server) en heeft minder overhead, betere controle en verfijning bij het alloceren van resources(CPU, disk, netwerk, geheugen) naar de virtuele machines. Er is ook een betere beveiliging en de virtuele machines zijn afgezonderd van elkaar. De afkorting ESX heeft geen betekenis maar zou vroeger gestaan hebben voor "Electric Sky eXclamation". De ESX 3.5 kernel is 48 MB groot.



Figuur 46: VMware ESX

- ◆ ESXi 3.5 is de nieuwste versie en dient niet geïnstalleerd te worden. We kopen hem op CD of USB stick en pluggen hem zo in in een host. ESXi start op van deze stick en kan verder alles wat de gewone ESX ook kan. Zijn footprint (grootte op de stick) is slechts 32MB.
- ◆ 2 van de functies van ESX die we belichten zijn de Virtual SMP & VMFS.
 - VMware Virtual SMP staat voor het feit dat de nieuwe ESX 3.5 in staat is om tot 4 virtuele CPUs per virtuele machine toe te kennen. Multithreaded applicaties kunnen op die manier gemakkelijk gevirtualiseerd worden.
 - VMFS (VMware File System): VMware heeft een eigen bestandssysteem ontwikkeld, de bestanden die hierop gebruikt worden zijn meestal super groot. (Een besturingssystemen is dikwijls enkele gigabytes groot). Een bestandssysteem met grote blok-groottes dringt zich op. Hierbij hebben we de keuze van blokken van 1MB tot 8MB, voor respectievelijk ondersteuning voor virtuele schijven van 256GB tot 2TB.

Verdere infrastructuur:

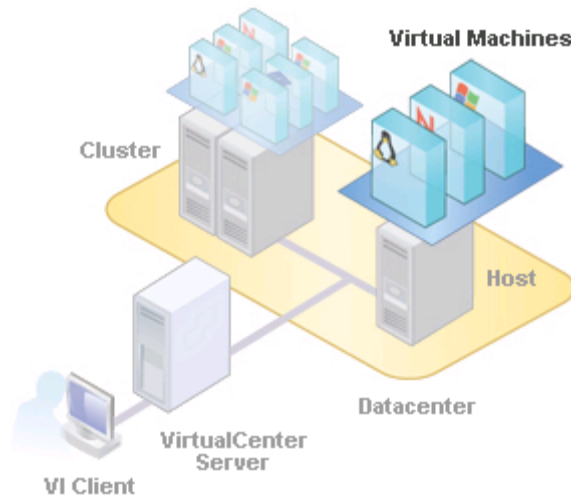


Figuur 47: Klassieke opstelling VMware Infrastructure

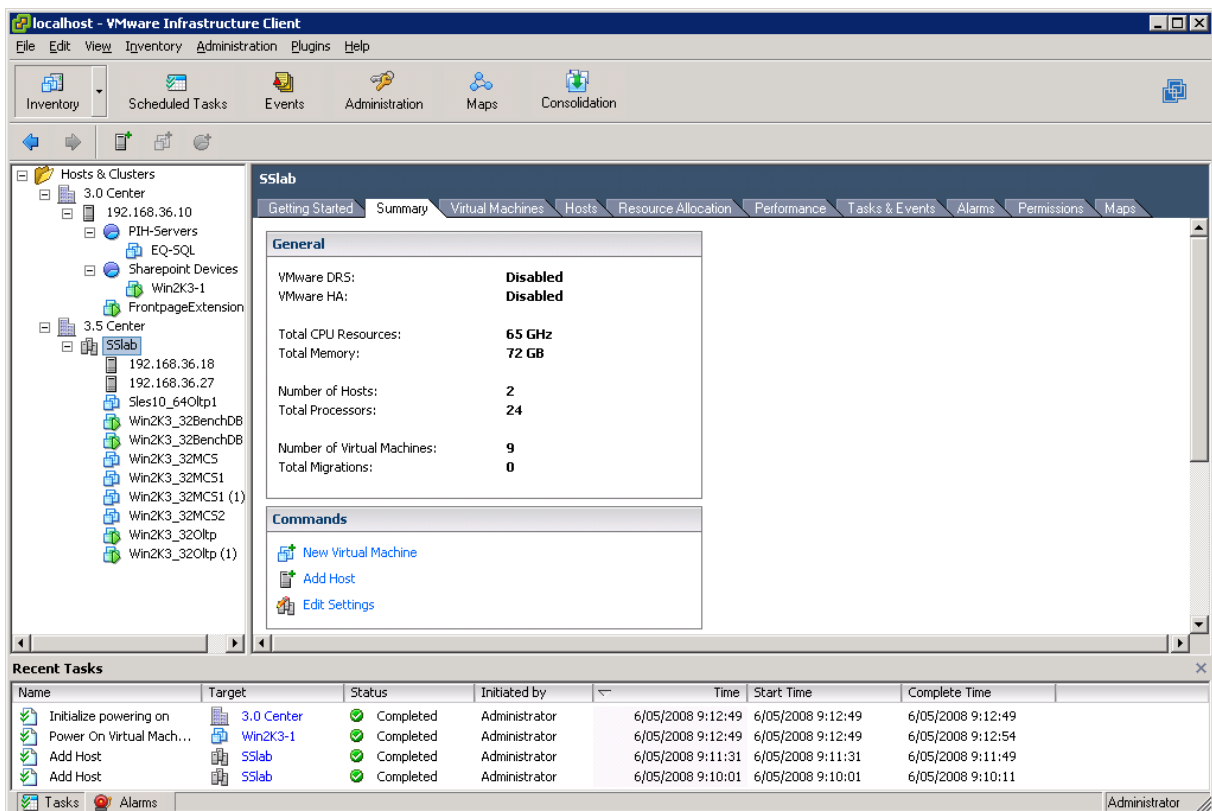
Als het motto van ESX is om servers te consolideren (aantal servers verlagen), zouden we dit echter niet meteen zeggen als we bekijken welke servers nodig zijn voor een volledige VMware Infrastructure. Gelukkig zijn de Virtualcenter server, license server & database op 1 pc te zetten en kan deze zelf virtueel draaien (bijv. op een ESX host).

Eén van de voordelen van virtualisatie is dat het behoorlijk eenvoudig is om virtuele machines over te zetten van de ene naar de andere host, dit heet migreren. De makkelijkste manier om dit te doen is het gebruiken van een Virtualcenter server. Deze geeft een overzicht van de verschillende hosts in het netwerk. Via een applicatie die draait op een client pc, bijvoorbeeld de laptop van een gebruiker, kunnen we verbinden met deze server en zo alles beheren. De nieuwste versie is Virtualcenter 2.5 en een database is nodig om de instellingen en gegevens van de verschillende hosts bij te houden. Deze kan apart van de Virtualcenter

geïnstalleerd worden, maar wordt meestal (net als de License Server) geïnstalleerd op dezelfde pc als de Virtualcenter pc.



Figuur 48: Virtualcenter werking



Figuur 49: Virtualcenter screenshot met cluster "SSlab"

Een cluster is een verzameling van 2 of meer hosts die door 1 virtualcenter beheerd worden. Op bovenstaande screenshot is zichtbaar dat voor 1 zo'n cluster alle CPUs, met hun verschillende snelheden, gewoon opgeteld worden, net zoals het geheugen. Er zijn hier verschillende datacenters zichtbaar, maar we kijken nu vooral naar het "3.5 Center" omdat hier meerdere hosts in zichtbaar zijn. "SSlab" is dan een cluster binnen dat datacenter. Onder

de 2 hosts (.18 & .27) zijn alle virtuele machines zichtbaar die staan op de 2 hosts en dus in de cluster aanwezig zijn.

Als VMware HA (High-Availability) ingeschakeld is voor die cluster, dan kan de Virtualcenter automatisch een virtuele machine van de ene host naar de andere migreren als 1 van de 2 zou falen. VMware DRS (Distributed Resource Scheduler) is iets geavanceerder, het maakt gebruik van een techniek die VMotion heet. Dit is eigenlijk migratie, maar dan zonder de virtuele machine uit te schakelen. De virtuele machine heeft zelden of nooit weet dat hij wordt overgeplaatst naar een andere host.

VMware DRS kan zien welke virtuele machine veel resources inneemt op een bepaalde host en kan bepalen dat deze virtuele machine beter op een andere host draait die meer van die resources vrij heeft. Hij zal dan automatisch een VMotion uitvoeren naar de host die dus de meeste resources vrij heeft (die het minste werk heeft op dat moment).

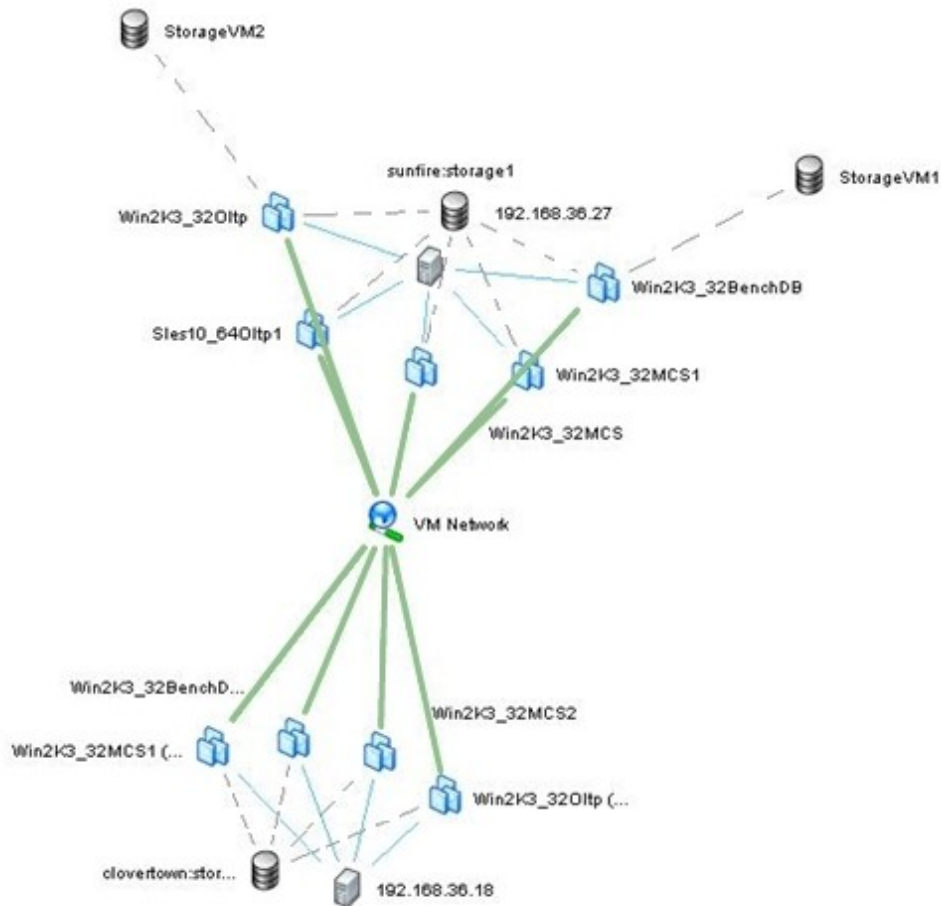
Het spreekt voor zich dat VMware HA & DRS enkel werkt als de bestanden van de virtuele machine (de virtuele harde schijf) op een fysieke schijf staat die voor iedere host toegankelijk is. De oplossing hiervoor is shared storage, maar dit werd reeds uitgebreid besproken in het eerste hoofdstuk.

Er zijn tabbladen om een overzicht van de virtuele machines of de hosts te krijgen, we kunnen tasks & events instellen. Deze laatste is bijvoorbeeld handig als de stroom uitvalt, de UPS geeft dan door aan het virtualcenter dat hij nog stroom kan leveren voor slechts enkele minuten. Dan kan de virtualcenter eerst alle virtuele machines proper afsluiten voor hij de host zelf afsluit. Bij het tabblad “Tasks & Events” kunnen we dan de volgorde instellen waarin hij de virtuele machines moet afsluiten. Als 1 VM een client is voor een andere die de server is, dan is het handig om eerst de client af te sluiten. Op dezelfde manier kunnen we werken bij het opstarten van de host: sommige virtuele machines mogen meteen opstarten samen met de host, andere best handmatig.

Het tabblad “Alarms” is dan weer handig voor als er problemen zijn met één van de VMs of één van de hosts, het Virtualcenter kan dan mails versturen naar de netwerkbeheerder of andere ingestelde activiteiten uitvoeren.

Het “Permissions”-tabblad dient voor het instellen van rechten op bepaalde VMs. In een bedrijf is het handig voor developers om enkele virtuele machines te kunnen beheren, maar andere productie-servers mogen voor hen dan niet toegankelijk zijn. We kunnen deze dan onderbrengen in een apart datacenter of aparte cluster en de rechten dan per datacenter of cluster anders instellen.

Het laatste tabblad geeft een visuele kaart weer van alle hosts, clusters, virtuele machines, opslag-schijven (LUNs) en hun onderlinge verbindingen.



Figuur 50: VMware Virtualcenter kaart van verschillende hosts

In bovenstaande afbeelding staan de dikke, groene lijnen voor netwerk-verbindingen, de blauwe lijnen duiden aan welke VMs op welke host draaien. De onderbroken, grijze lijnen geven weer op welke LUN de harde schijf (of schijven) van een VM staat. In bovenstaand voorbeeld zijn er 2 hosts die elk met hetzelfde LAN-netwerk verbonden zijn.

4.2. Concrete installatie van een ESX Host

De installatie van ESX verklaart eigenlijk meestal zichzelf. CD insteken en enkele keren op Next drukken volstaat om in zeer korte tijd (<30min) een ESX host draaiende te hebben. Maar daarom is het nog niet altijd duidelijk wat we doen. Vooral bij het partitioneren van een ESX System kunnen er soms wel onduidelijkheden zijn.



Figuur 51: ESX-setup: start-scherm

Bij het start-scherm kunnen we gerust de grafische mode kiezen door op [Enter] te drukken.

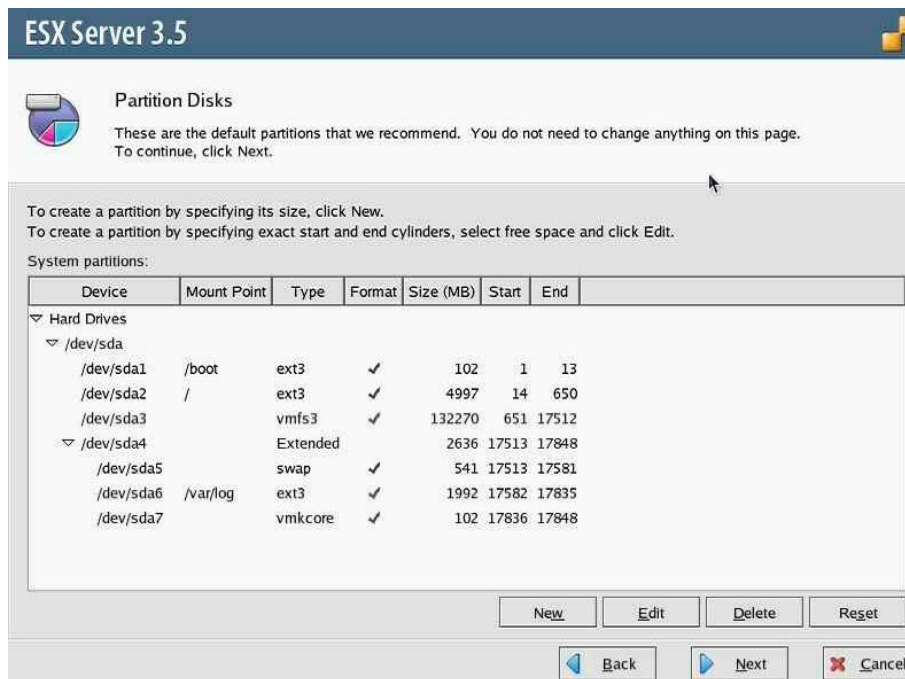
Via het scherm erna kunnen we de CD testen op eventuele fouten, maar als we dit ooit al gedaan hebben kunnen we dit overslaan.

Na een welkomscherm komen we bij de toetsenbord-keuze, we gaan hier helemaal naar boven in de lijst (drukken op de Home toets werkt misschien het best) als we een azerty-toetsenbord willen. We kunnen hier uiteraard iets anders instellen, indien gewenst.

Voor de muis mogen we verdergaan zonder kijken, gewoon op Next drukken is voldoende.

Deze toetsenbord- en muis-instellingen zijn enkel van nut als we via de lokale ESX Console willen werken, dus enkel tijdens de installatie. Daarna gebeuren alle configuraties via een client en worden deze instellingen van de client overgenomen.

Het volgende scherm laat toe om de harde schijf te kiezen waar we ESX op willen installeren, we kunnen ook kiezen voor geavanceerd (Advanced), maar voorlopig laten we dit zo.



Figuur 52: ESX-setup: Partitionering

Het volgende scherm ziet er iets ingewikkelder uit, een kleine verduidelijking is op zijn plaats. Hier kunnen we de volledige partitie-tabel overzien en eventueel aanpassen.

ESX kan enkel met MSDOS-partitie-tabellen werken, dus geen GPT. Dit heeft als nadeel dat deze geen partities van meer dan 2TB kan aanspreken en dat er maar 4 primaire partities kunnen ingesteld worden. ESX heeft echter een 6-tal partities nodig en maakt daarom eerst 3 primaire partities aan waarna hij van de 4^{de} partitie een extended partitie maakt om de overige 3 primaire op te zetten.

Als eerste is er uiteraard een boot-partitie nodig, de MBR verwijst naar deze partitie als de locatie van de bootloader. Hierin zitten tevens de kernel-images en de service console image, om te kunnen starten in "Troubleshoot Mode". We kunnen deze groter maken, om plaats te hebben voor eventuele ESX-updates. Maar in de praktijk is het even makkelijk en veel veiliger om een volledige her-installatie te doen.

De tweede partitie is dan de root-partitie van het Linux-OS en omvat alle gegevens die ESX nodig heeft om te werken. Dit is dus de gestripte Linux waar ESX mee werkt en omvat tevens de webserver en zijn bestanden.

De derde is een partitie met het VMware File System (VMFS) erop, zoals eerder besproken is dit de plaats waar de schijf-images van de virtuele machines opkomen. Deze is dus uiteraard zo groot mogelijk. Maar kan later uitgebreid worden (via SAN, NAS of DAS).

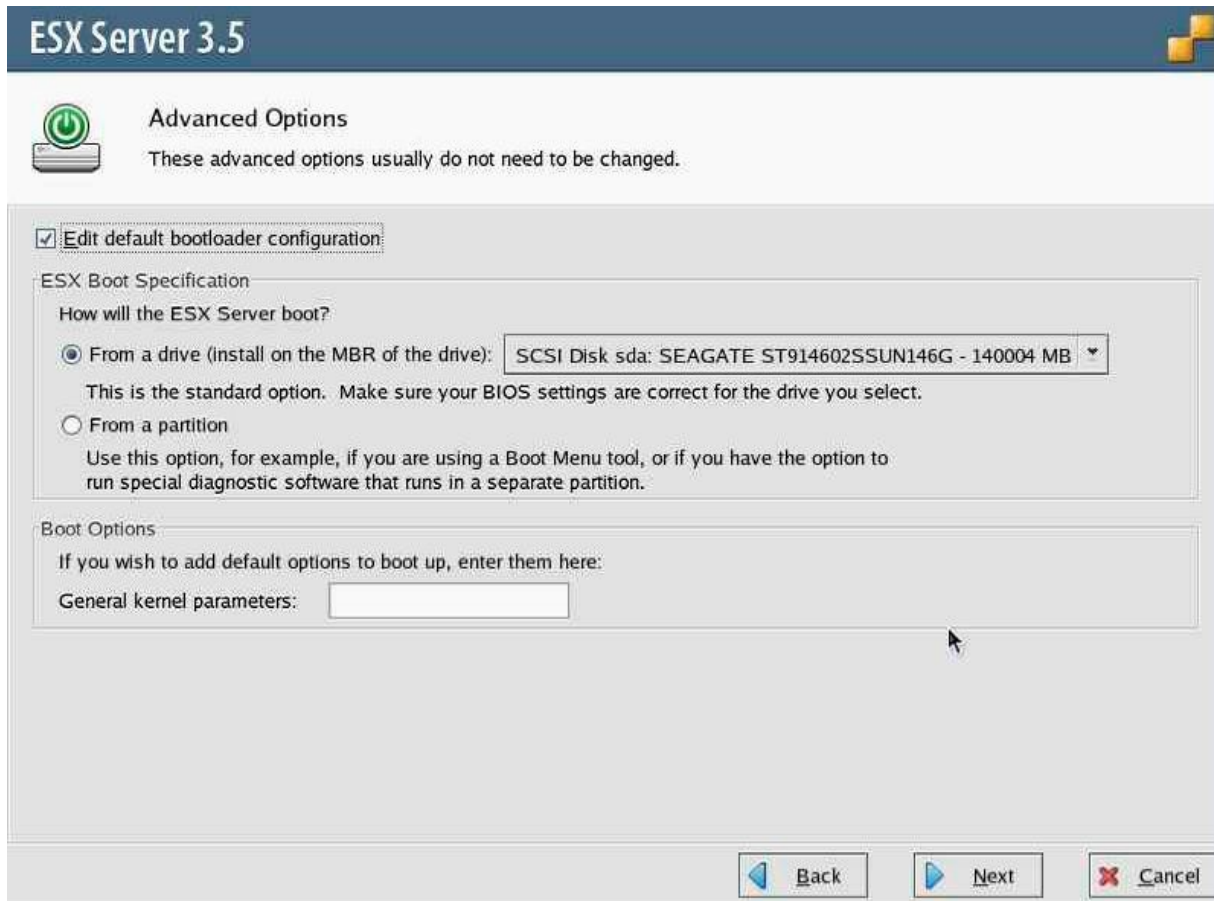
De 4^{de} partitie (extended partitie) is dan de swap, voor eventueel geheugen uit te wisselen als de RAM niet voldoende groot is. Dit is niet van toepassing op de virtuele machines, maar enkel op het beheer van de ESX zelf en dient dus geen veelvoud van het fysieke geheugen te zijn.

De 5^{de} partitie dient uitsluitend voor de log-files van ESX.

De 6^{de} en laatste partitie dient om volledige kopieën op te slaan van de ESX Core. Als er iets

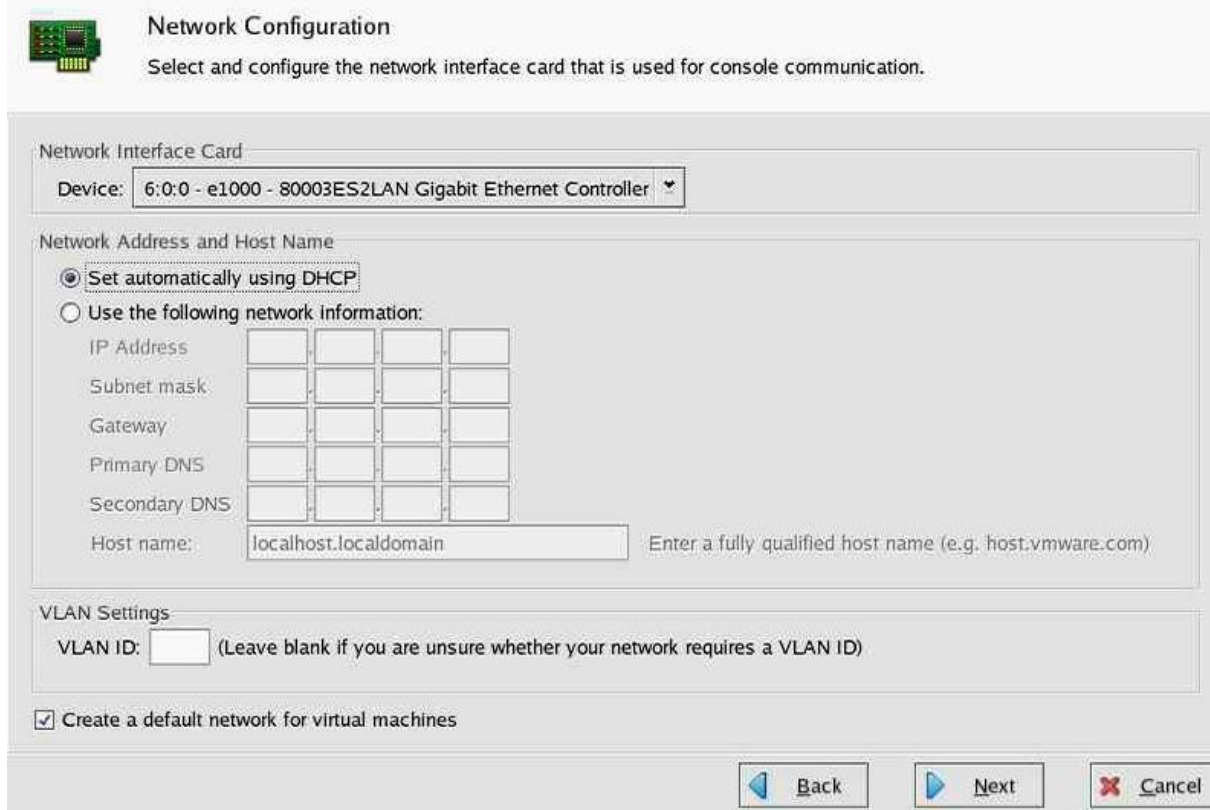
mis is met de ESX kunnen we deze kopieën doorsturen naar VMware voor evaluatie. Als we deze partitie niet hebben biedt VMware geen ondersteuning.

Daarna volgt het scherm met de bootloader configuratie, dit kunnen we ook gerust standaard laten.



Figuur 53: ESX-setup: Bootloader

Ook het volgende scherm is zeer belangrijk. In dit scherm wordt de netwerk-configuratie ingesteld. Hierover moet zeker worden nagedacht want dit is na installatie zeer moeilijk aan te passen. Zonder (ondersteunde) netwerkkaart wil de setup van ESX zelfs niet starten, want het aanmaken en beheren van Virtuele Machines gebeurt via een webinterface of web-client. Standaard staat dit venster op DHCP venster, maar via een vast IP is het eenvoudiger om de server te bereiken en is het mogelijk om een Host name toe te kennen. We kunnen ook een VLAN-id aanmaken (handig als we verschillende ESX servers in aparte VLANs zetten) en standaard creëren we best ook een netwerk voor de virtuele machines. Zodat we dit (virtueel) netwerk kunnen toekennen aan een VM en deze op het netwerk kan.



Network Configuration
Select and configure the network interface card that is used for console communication.

Network Interface Card
Device: 6:0:0 - e1000 - 80003ES2LAN Gigabit Ethernet Controller

Network Address and Host Name
 Set automatically using DHCP
 Use the following network information:
 IP Address: [][][][]
 Subnet mask: [][][][]
 Gateway: [][][][]
 Primary DNS: [][][][]
 Secondary DNS: [][][][]
 Host name: localhost.localdomain Enter a fully qualified host name (e.g. host.vmware.com)

VLAN Settings
VLAN ID: [] (Leave blank if you are unsure whether your network requires a VLAN ID)

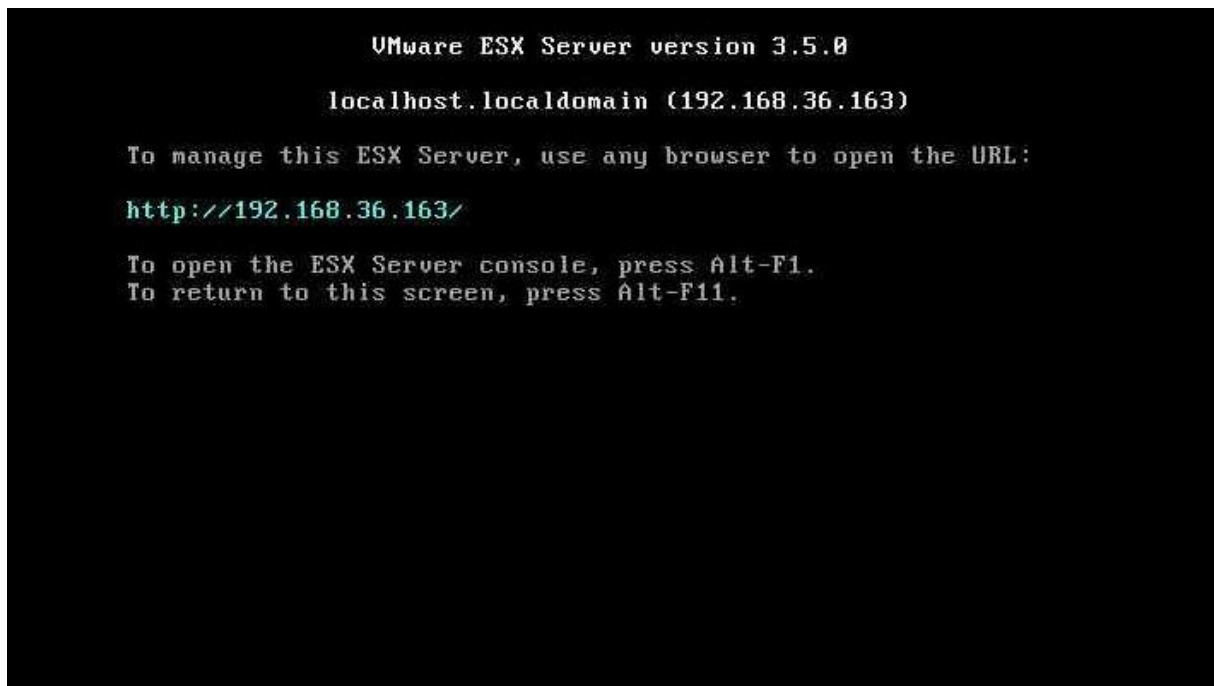
Create a default network for virtual machines

Back Next Cancel

Figuur 54: ESX-setup: Netwerkkonfiguratie

Na de tijdszone gekozen te hebben krijgen we nog een installatie-samenvatting en begint de installatie. Deze duurt (afhankelijk van de server-uitrusting) 15 tot 30 minuten.

Na een herstart, en als de schijf niet verbonden is met een SATA MCP55 Controller, is de machine klaar en het enige dat nog zichtbaar is op de server een zwart scherm met IP-adres. Hier kunnen we via de toetsencombinatie *Alt-F1*, toch nog in een Linux-console terecht komen waar we bijv. SSH toegang kunnen aanzetten, bepaalde administratieve zaken uitvoeren of geavanceerde monitoring doen (bv *ESXTOP*).



Figuur 55: ESX-setup: Gestart scherm

4.3. Configuratie, optimalisatie van een ESX-server

Het is mogelijk om ESX 3.5 60 dagen te gebruiken zonder registratie-sleutel. Anderzijds kunnen we deze op 3 manieren registreren. Door een sleutel in te geven, door een volledig sleutel-bestand op te geven of door het adres van een licentie-server in te geven. Let er wel op dat de licenties van ESX per socket worden verdeeld, en niet per CPU. Voor 2 Quad-core processors zijn er maar 2 licentie's nodig, terwijl 4 Dual-Core processors 4 licenties vereisen. Zoals eerder aangehaald gebeurt alle management via de VMware Infrastructure Client 2.5 (versie 2.0 is voor ESX 3.0).



Figuur 56: VMware Infrastructure Client

We kunnen met deze client meteen verbinden naar de client (door zijn IP adres in te geven) en daar kunnen we reeds veel lokale dingen instellen. Zoals zijn storage management (verbinding maken met een iSCSI of NFS Server). Echter, om gebruik te kunnen maken van geavanceerde (& dure) features zoals VMotion, DRS of HA zijn we verplicht om elke ESX server te beheren onder één algemene tool; het Virtual Center. Dit is dus een aparte, kleinere server met de Virtual Center software op. Deze server kan ook dubbel als database- & license server.

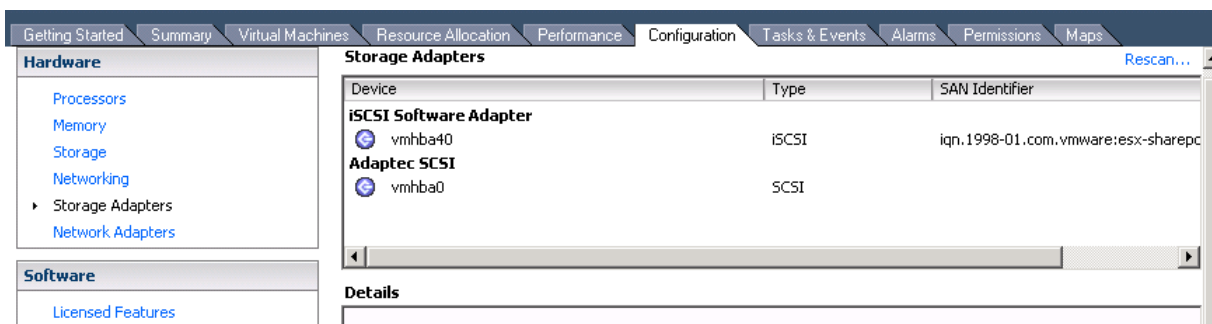
We kunnen met onze VMware Infrastructure Client verbinden naar deze Virtual Center software en kunnen daar hosts toevoegen aan een zogenaamde Resource Pool. Dit is een verzameling van hosts die als één groot datacenter kunnen bekeken worden. Met aparte

restricties, maar vooral waarbinnen een tool als DRS zijn werk mag doen. DRS zal bijvoorbeeld geen machines van de ene Resource Pool naar de andere verplaatsen.

4.3.1. Virtuele schijven op aparte schijf

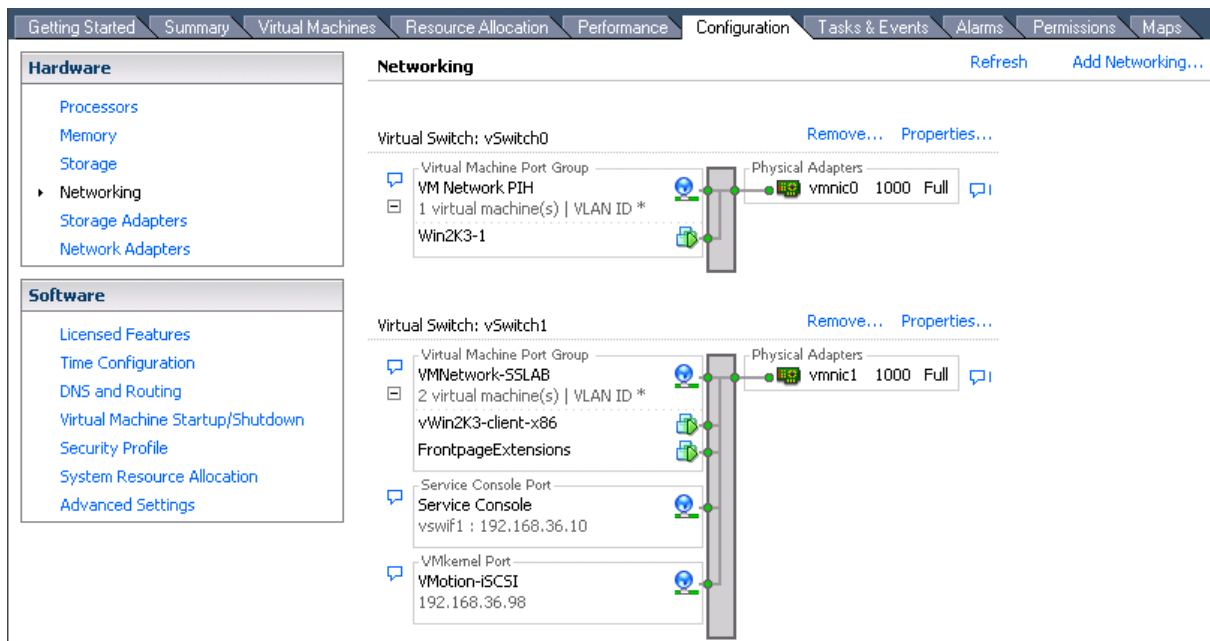
Een virtuele machine kan gebruik maken van VMotion. Of er is een aparte database-schijf nodig. Of het is een bestands-server die terabytes aan data moet verzetten & gebruiken. Allemaal redenen om het opslag-gedeelte of de volledige virtuele schijf te huizen op een aparte fysieke schijf, ofwel op een SAN (Fibre Channel of iSCSI) ofwel een NAS (oa NFS). De manier om dit te doen, in het geval van ESX, is door het creëren van datastores. Dit zijn stations (volumes of LUNs) die beschikbaar zijn voor één of meerdere ESX hosts om hun virtuele schijven op te zetten. Bij installatie van ESX kunnen we er reeds aanmaken, maar de meeste worden na installatie opgezet.

Iedere hardware adapter (onboard SATA-controller of PCI-E SAS-controller ...) is standaard zichtbaar onder het menu “Storage Adapters”, eventueel na een druk op de knop “Rescan...”. Enkel volumes verbonden met deze zichtbare adapters kunnen gebruikt worden als datastorage. Als we ze willen partitioneren & formatteren (VMFS) dan dienen we naar het menu “Storage” te gaan en op de knop “Add storage” te drukken, dan wordt daar het volume gedetecteerd en kunnen we verder gaan.



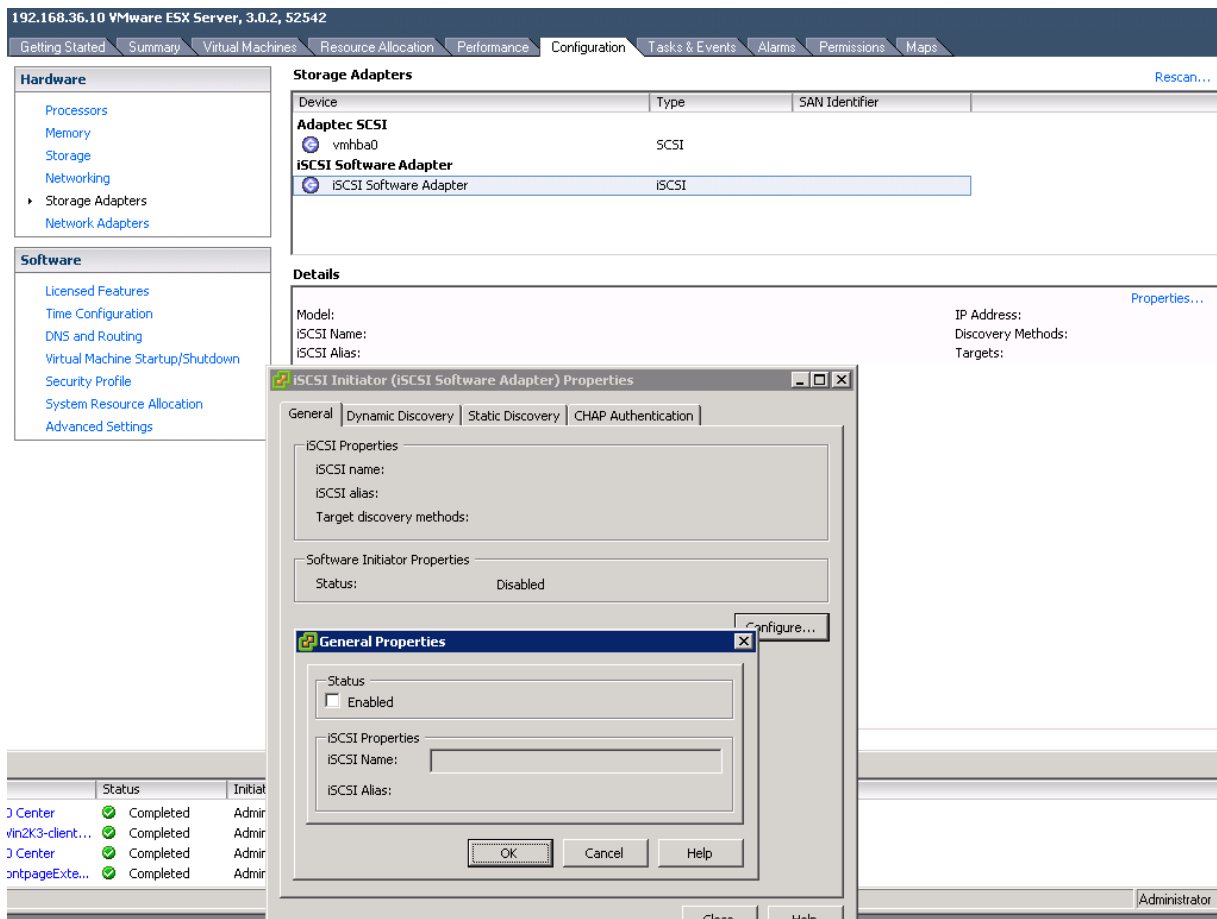
Figuur 57: ESX-optimalisatie: Storage Adapters

Als we een SAN of NAS willen gebruiken als datastorage dienen we hiervoor een aparte (virtuele) netwerkkaart aan te maken, dit kan onder het menu “Networking”.



Figuur 58: ESX-optimalisatie: Networking

ESX werkt met virtuele switches die als poort een fysieke netwerkkaart kunnen hebben. In principe kunnen we een onbeperkt aantal virtuele switches aanmaken. Meestal wordt er gewerkt met 1 switch per fysieke poort. Bij de switch waar we iSCSI op willen laten gebeuren kunnen we de eigenschappen bekijken via de knop “Properties...”. Daar kunnen we via de knop “Add” een VMkernel verbinding toevoegen, deze krijgt een eigen IP-adres en kan gebruikt worden voor o.a. iSCSI & NFS. We geven hem een label en stellen best zijn IP-adres vast in. Daarna kunnen we bij “Storage Adapters” de iSCSI Software Adapter selecteren en zijn eigenschappen opvragen. Via de de knop “Configure” vinken we “Enabled” aan en dan krijgt de ESX server automatisch de nodige IQN.



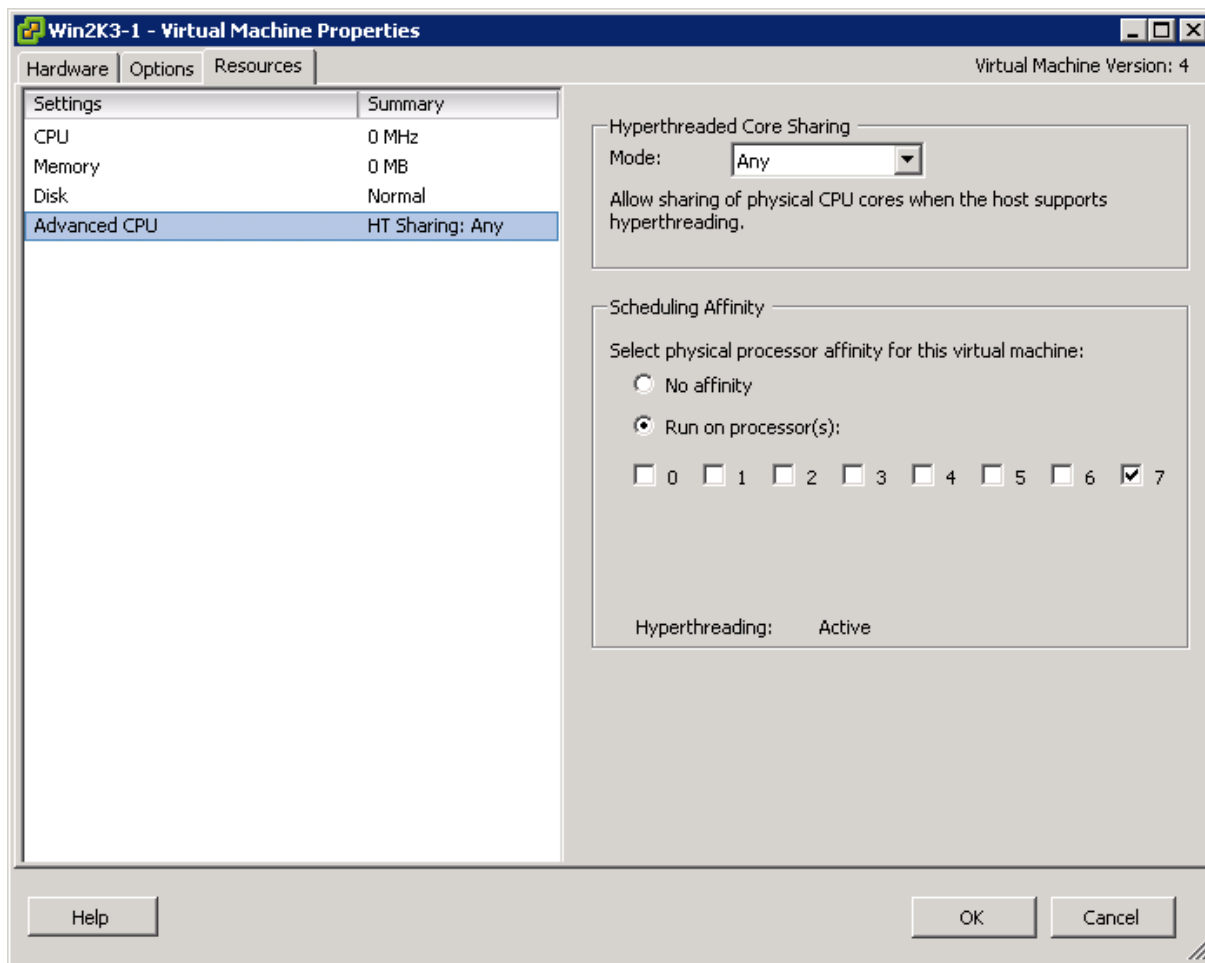
Figuur 59: ESX-optimalisatie: iSCSI Settings

Bij het tabblad “Dynamic Discovery” geven we dan de ip-adressen in van de iSCSI servers. Dan kunnen we bij “Storage” een nieuw volume toevoegen met de knop “Add Storage”. Alsook NFS volumes kunnen op deze manier toegevoegd worden.

4.3.2. Processor affiniteit

Een optie die we echter in nog geen enkele andere virtualisatie-oplossing waren tegengekomen, is het instellen van de processor affiniteit. Affiniteit instellen wil zeggen dat we een proces, taak of virtuele machine uitvoeren op een bepaald aantal CPUs. Deze optie is ook terug te vinden in Windows taakbeheer.

Standaard zal ESX alles wat een VM wil laten uitvoeren door de CPU, gelijkmatig verdelen over alle beschikbare CPUs. Dit verdelen is een behoorlijk zware last op de schouders van de Hypervisor en bijgevolg is het aan te raden om een VM vast te pinnen aan een bepaald aantal CPU kernen. Als de VM dus 2 vCPUs toegewezen kreeg, pinnen we hem vast aan 2 CPU kernen. Om dit te doen gaan we naar de settings van de VM en kiezen dan voor het laatste tabblad “Resources”, daar de optie “Advanced CPU” en dan klikken op “Run on processor(s)”.



Figuur 60: ESX-optimalisatie: Processor affiniteit

4.3.3. VMware Tools

Ook op software gebied van de VM kunnen we iets aanpassen, namelijk door installatie van de VMware Tools. Dit zijn speciale drivers en configuratie-software geschreven door VMware en beschikbaar voor zowel Windows als Linux. Het installeert oa speciale drivers voor de virtuele netwerkkaart en grafische kaart. De VMware Tools kunnen ook zorgen voor tijdssynchronisatie tussen de ESX server en de virtuele machine. Ook kunnen ze speciale ESX scripts uitvoeren, zoals een defragmentatie op bepaalde tijdstippen of het gast-systeem keurig afsluiten als de ESX afgesloten wordt.

Om zeker te zijn dat alle aspecten van de VMware Tools goed geïnstalleerd zijn is het aan te raden deze te compileren op ons 64-bit SLES-systeem. Hiervoor hebben we de kernel-source nodig die we installeren met het commando: `# yast -i kernel-source`. We zorgen er best voor dat de SLES installatie dvd gekoppeld is met de VM. Daarna kunnen we gemakkelijk de VMware Tools compileren door de bronbestanden (tar-bestand) uit te pakken (`# tar -xZf VMwaretools-3.5.0-64607.tar.gz`). We merken dat er voor ons een compilatie-script is geschreven in de scripting taal: Perl. We typen nu gewoon `./vmware-install.pl` en drukken telkens op *Enter* om de standaard waarden te aanvaarden.

4.3.4. Partitie alignering

Zoals eerder in het Shared Storage-hoofdstuk 1.3 is aangehaald, kan het aligneren van partities een grote prestatie-impact hebben bij schijf intensieve applicaties. Dit blijft echter niet beperkt tot Windows partities; ook Linux partities hebben baat bij het instellen van een offset. Echter (net zoals bij Windows), dit is niet mogelijk via de gebruikersinterface die de VMware Infrastructure Client biedt. We kunnen ook VMFS-partities aanmaken via de console. Hiervoor zetten we een SSH-sessie op met de ESX-server in kwestie, of we werken rechtstreeks in de ESX-console.

OPGELET: standaard is SSH-toegang niet toegelaten op een ESX-server, hiervoor moeten we eerst lokaal de “PermitRootLogin” parameter op “yes” zetten in het bestand `/etc/ssh/sshd_config`.

Een stappenplan hiervoor is te vinden in Appendix D: ESX Troubleshooting

5. Xen, kostenloze virtualisatie

Xen was oorspronkelijk een project binnen de Universiteit van Cambridge, waar het ontwikkeld werd door de mensen van het bedrijf XenSource, Inc. Xen werd gereleased in 2003 onder een GNU Public License, wat betekent dat het vrije software is. In oktober 2007 werd Xen echter overgekocht door Citrix en verhuisde op dat moment naar xen.org. Verder is XenSource wel nog bezig aan het ontwikkelen van een compatibiliteit laag voor Microsofts Hyper-V zodat gast systemen die draaien onder Xen meteen kunnen werken onder Hyper-V ook.

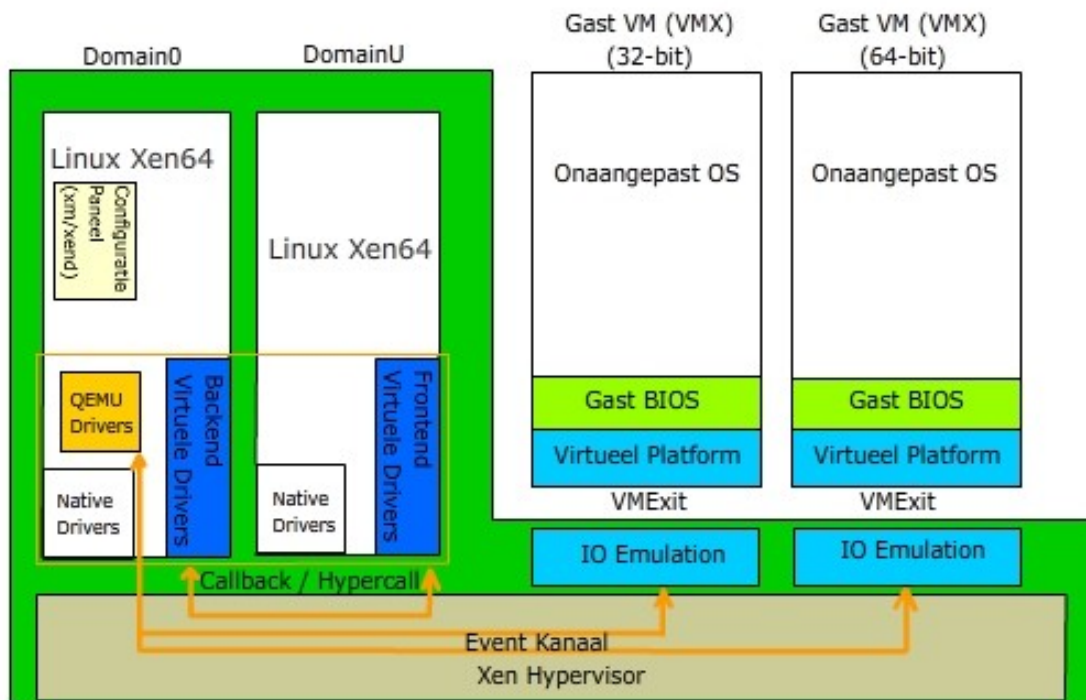
Xen werd reeds standaard ingebouwd in diverse Linux distributies zoals Novell SLES10, Red Hat's Enterprise Linux 5 of Fedora 7.

5.1. Werking

Zoals reeds uitgelegd bij paravirtualisatie werkt Xen met een Hypervisor die zich tussen de virtuele machines en de hardware bevindt. Deze kan echter niet autonoom werken, deze werkt samen met de speciale VM0. Virtuele machines heten onder Xen domains. Domain0, dom0 of VM0 is een speciaal domain, dit is namelijk de virtuele machine die het beheer doet. Deze privileged machine heeft speciale mogelijkheden & rechten en bevat vooral alle native drivers die de andere geparavirtualiseerde domains kunnen aanspreken (zie afbeelding bij paravirtualisatie).

Verder heeft Xen ook nog domU's die dan minder geprivilegeerd zijn, ook hier zijn er weer 2 soorten. De geparavirtualiseerde domains en, vanaf Xen 3.0, de HVMs (Hardware Virtual Machine). Geparavirtualiseerde VMs hebben een frontend interface die doorlinkt naar de native apparaat drivers die geïnstalleerd staan in dom0.

HVMs zijn VMs die in Full Virtualization draaien, dit is echter enkel mogelijk met behulp van Hardware Virtualization, dus met Intel-VT-x of AMD-V CPUs.



Figuur 61: XEN HVM: 32bit- & 64bit-ondersteuning

QEMU is een aparte emulator die met behulp van, het eerder besproken, Binary Translation hardware kan afschermen en virtualiseren voor besturingssystemen die in Full Virtualization draaien. Dit is dus een tweede set drivers die in de dom0 dient aanwezig te zijn. Dit zijn echter Linux-drivers die eventueel een IO Emulatie nodig hebben om aangesproken te kunnen worden door een Windows systeem.

Deze hele vertaalslag zorgt ervoor dat Full Virtualization een zwaar prestatieverlies teweegbrengt onder Xen en bijgevolg is het beter enkel Xen-enabled besturingssystemen te gebruiken. Enkele besturingssystemen die reeds geparavirtualiseerd zijn, zijn te zien in volgende lijst:

- ◆ Linux, vanaf kernel 2.6.23, patches voor eerdere versies bestaan ook
- ◆ NetBSD, NetBSD 2.0 ondersteund Xen 1.2, NetBSD 3.0 Xen 2.0 & NetBSD 3.1 Xen 3.0
- ◆ OpenBSD
- ◆ FreeBSD, in beperkte mate
- ◆ OpenSolaris
- ◆ NetWare

5.2. Xen installatie

Zoals eerder aangehaald is Xen voorgecompileerd in diverse Linux distributies zoals Red Hat Enterprise Server & Novell Suse Linux Enterprise Server 10.

5.2.1. Algemeen

Als Xen geïnstalleerd staat op een distributie kunnen we bij het starten kiezen tussen Xen en niet-Xen. De standaard native distributie wordt m.a.w. niet overschreven.

Dit kunnen we het duidelijkst zien in de (grub-)bootloader. Er is een nieuwe entry genaamd Kernel-2.6.18.2-34-Xen. Met als grootste wijziging dat de te laden kernel /boot/xen.gz heet en dat de aangepaste SLES-kernel als module meegegeven wordt. Deze SLES-kernel wordt dan uiteindelijk een driver-domain (Dom0 met native drivers).

```
title Kernel-2.6.18.2-34-xen
    root (hd0,2)
    kernel /boot/xen.gz
    module /boot/vmlinuz-2.6.18.2-34-xen root=/dev/hda3 vga=0x317
                                                resume=/dev/hda4 splash=silent showopts
    module /boot/initrd-2.6.18.2-34-xen
```

Dit is het stappenplan als in de bootloader gekozen wordt voor het Xen-enabled systeem.

1. De Xen Hypervisor wordt geladen als eerste softwarelaag bovenop de hardware.
2. De geparavirtualiseerde dom0 Linux-kernel wordt bovenop de hypervisor ingeladen.
3. De Xen Daemon (genaamd xend) wordt gestart in dom0. Deze daemon staat in voor het starten en stoppen van domains. Alsook het beheer van het virtuele netwerk voor de domains. Dit gebeurt aan de hand van configureerbare scripts.

De domains beheren gebeurt ook via commando's die Xen Daemon aanbiedt. Per distributie zijn er diverse grafische tools om hetzelfde te bereiken. Maar iedere versie heeft standaard een Xen commando die hetzelfde kan doen. Dit commando heet *xm* en staat voor Xen Manager. Afhankelijk van de meegegeven parameter kunnen we andere dingen opvragen, maar het is simpeler om in de *xm shell* te gaan via het *xm shell* commando. Op die manier kunnen we het woord *xm* bij de volgende commando's telkens weglaten.

Zo is er *xm info* om alle Xen informatie te zien, *xm top* (of *xentop*) toont de gebruikte resources door de domains. Om naar de console van een geparavirtualiseerd domain te gaan (ook als er geen netwerk is) kunnen we *xm console \$id* gebruiken, deze kan verbroken worden met *ctrl+J*. Dit werkt echter niet voor HVMs. Het \$id is een identificatienummer per domain en is te zien via *xm list*. Dit laatste commando toont ook andere informatiezo als het aantal toegekende CPUs en zijn status, maar enkel van de draaiende domains.

Een domain aanmaken gebeurt via het *xm create \$configuratiebestand* commando, waarna hij een xml-bestand met alle instellingen aanmaakt en het domain start. Dit nieuw domain moet ook nog toegevoegd worden aan de Xen manager met *xm new \$naam*, want anders is het niet opstartbaar met *xm start \$naam*.

Een bestaand domain starten kan via het *xm start \$naam* commando. Afsluiten kan met *xm shutdown \$naam*, *xm pause \$naam* & *xm unpause \$naam* dienen om het domain te pauzeren en respectievelijk te hervatten. Met *xm destroy \$naam* kunnen we een volledig domain in 1 keer afsluiten, dit is niet mogelijk met dom0.

Je kunt ook een domain-status opslaan, zodat dit afgesloten wordt en later opnieuw opgestart vanuit dezelfde status. Dit kan met *xm save \$id \$file* en *xm restore \$file*.

In plaats met identificatie nummers te werken (\$id) is het ook mogelijk om de naam van het configuratie-bestand te gebruiken.

Het opzetten van een domain (VM) gebeurt onder Xen volledig via configuratiebestanden. De locatie van deze bestanden kan lichtjes verschillen van distributie tot distributie maar meestal zijn ze te vinden op volgende locaties(Novell Xen): */etc/xen, /etc/lib/xen/boot & /var/lib/xen*. In de eerste map bevinden zich de instellingen van de Xen Daemon en instellingen van de domains. In */etc/lib/xen/boot* bevinden zich dan de start-scripts voor de diverse domains en in */var/lib/xen* zitten de eigenlijke harde schijf bestanden van de domains plus de database van de Xen Daemon die de status bijhoudt van elk domain.

Een configuratie-bestand dat een paravirtueel domain beschrijft, heet een python-script en kan er typisch als volgt uitzien:

```
disk = [ 'file:/var/lib/xen/images/sles10-vm5/hda,hda,w', 'phy:/dev/hdb,hdb,r' ]
memory = 512
vcpus = 1
builder = 'Linux'
name = 'sles10-vm5'
vif = [ 'mac=00:16:3e:5c:96:a8' ]
on_poweroff = 'destroy'
on_reboot = 'restart'
on_crash = 'restart'
extra = 'TERM=xterm'
bootloader = '/usr/lib/xen/boot/domUloader.py'
bootentry = 'hda2:/boot/vmlinuz-xen,/boot/initrd-xen'
```

De eerste parameter bevat de schijven van de virtuele machine. Er zijn 3 soorten: bestanden (virtuele schijf bestanden), fysieke partities of LVM (Logical Volume Manager) schijven. We dienen binnen de enkele aanhalingstekens eerst de soort & locatie van het fysieke gedeelte in te geven. Daarna (met een komma gescheiden) de naam van het apparaat binnen het virtuele domain en tenslotte ook de mode, r voor read-only en w voor schrijf-toegang. De tweede parameter is de grootte van het geheugen. De derde is het aantal virtuele CPUs. Bij builder kunnen we opgeven of het om full- of paravirtualisatie gaat, de optie's zijn "Linux" of "hvm". Bij "name" hoort de naam en vif dient om het virtuele netwerk interface in te stellen. Als we vif [''] invullen is dit standaard en wordt dit later automatisch aangevuld, maar we kunnen ook een mac-adres meegeven zoals hierboven. Met on_poweroff, on_reboot & on_crash kunnen we het gedrag instellen als het domain respectievelijk afsluit, herstart of crasht.

Met het extra-commando kunnen we parameters meegeven aan de kernel, zoals waar de terminal moet plaatsvinden. Tot slot geven we nog mee op welke locaties de bootloader & bootentry te vinden zijn. Let wel, dit configuratiebestand kunnen we aanpassen naargelang wat we willen doen. Als we het besturingssysteem van het domain nog moeten installeren zullen we bijvoorbeeld een fysiek cd-rom station als schijf toevoegen en als extra parameters bij de kernel stellen we de textmode en de terminal naar xterm in.

5.2.2. Driver domains

Er bestaan onder Xen geprivilegerde virtuele machines. Die dus native drivers aan boord hebben om op die manier rechtstreeks hardware te kunnen aanspreken. Hier even een concreet voorbeeld van hoe we dit aanpakken voor zo'n domU dat als enige de geluidskaart moet kunnen aanspreken.

- ◆ We zoeken het id op van het PCI-apparaat dat we willen uitsluiten voor alle andere domain (o.a. het Xen beheer domain dom0). Dit kan met `lspci`, we vinden zo bijvoorbeeld:
00:1b.0 Audio device: Intel Corporation 82801G (ICH7 Family High Definition Audio Controller (rev01))
Hier is het PCI-id dus *00:1b.0*.
- ◆ Dit dienen we nu toe te voegen in de (grub-)bootloader op de eerste module lijn, dit geeft dan zo een regel:
*module /boot/vmlinuz-2.6.18.2-34-xen root=/dev/hda3 vga=0x317
resume=/dev/hda4 splash=silent showopts pciback.hide(00:1b.0)*
Meerdere apparaten verbergen kan door het `pciback`-commando te herhalen na het laatste haakje.
- ◆ Tot slot dienen we dit apparaat nog toe te voegen in het Xen-configuratiebestand (`/etc/xen/vm`). We maken gewoon een nieuwe regel en typen `pci=['00:1b.0']`.

5.2.3. Een concrete installatie (onder SUSE Linux Enterprise Server 10 SP1)

Suse staat al jaren bekend om zijn gecentraliseerd systeembeheer dat als naam YaST (Yet another Setup Tool) meekreeg. Ook via deze tool kunnen we zeer makkelijk een native systeem omvormen naar een Xen systeem. Novell noemt een Xen-systeem een VM Server.

Na het opstarten van YaST kunnen we Xen installeren via `Virtualization > "Install Hypervisor & Tools"`. Daarna dienen we de computer te herstarten, zodat de Xen Hypervisor-kernel ingeladen wordt en niet de standaard SLES Linux kernel.

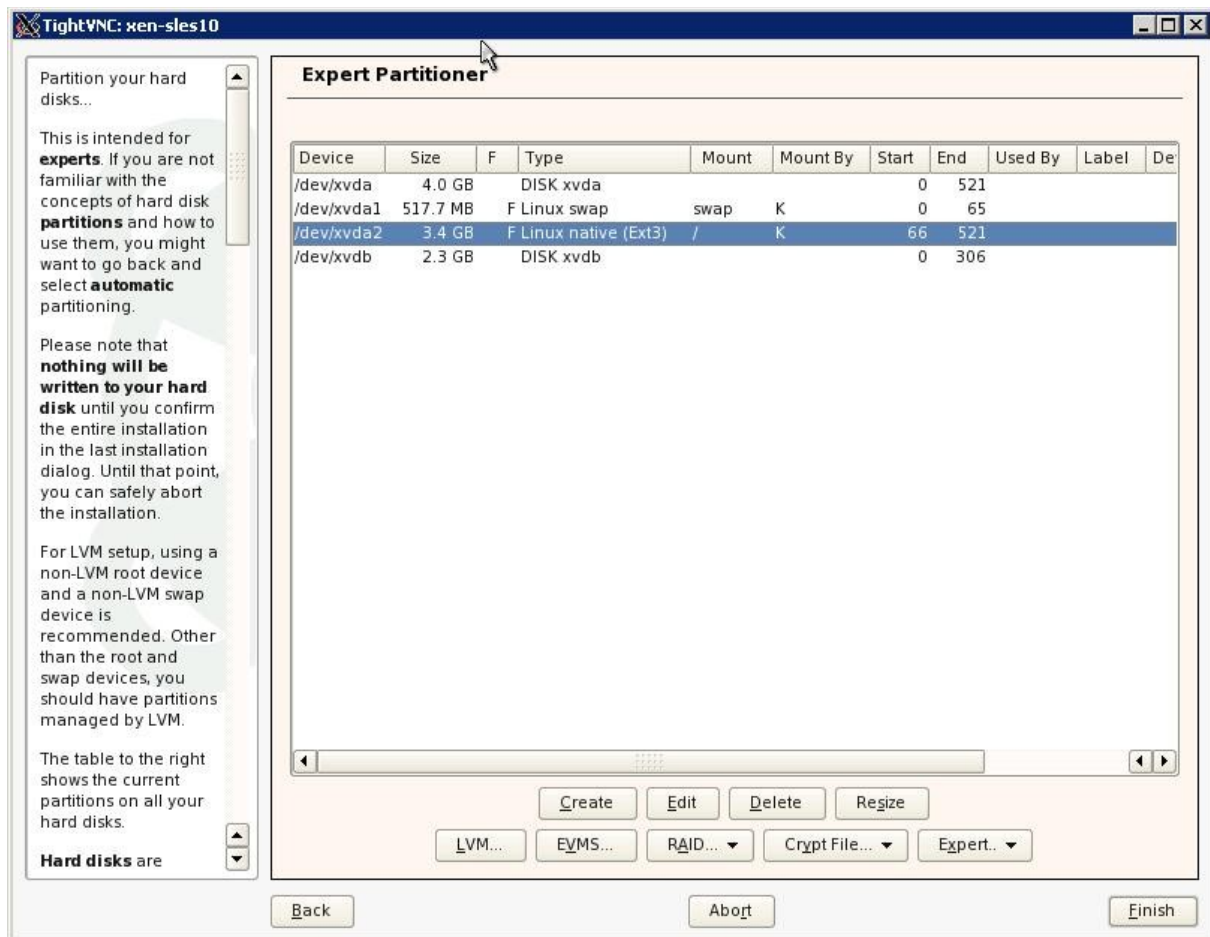
Het beheer van DomU's is het simpelst onder YaST maar kan ook via de eerder aangehaald `xm` command line tool.

Er zijn 3 opties om een virtuele machine op te slaan. Er is bestandsniveau, daarmee maken we een file-image aan en hangen deze aan het domain. In het configuratiebestand kunnen we dit met `file:/var/lib/xen/images/vm1` binnen de disk-parameter. Ten tweede zijn er de logische volumes van Linux, dit is een Linux-laag bovenop de partitie waarmee we dynamische volumes kunnen aanmaken die zeer flexibel zijn (vergelijkbaar met de dynamische volumes van windows). Deze kunnen met het `lvcreate`-commando aangemaakt worden en in het configuratiebestand is de juiste regel dan: `phy:/dev/volume/vol1` binnen de disk-parameter.

Ten derde kunnen we ook een reeds bestaande fysieke partitie mounten aan een domain, de regel is dan eerder als volgt: `phy:/dev/sda1` binnen de disk-parameter.

Via command line is het met de juiste Linux kennis dan mogelijk om een bestaand SLES10 besturingssysteem te kopiëren naar de juiste schijf-image en de juiste startup-instellingen in te stellen. Dit kan met `dd` of met `debootstrap`.

Het is echter eenvoudiger via de grafische YaST tool, bij Virtualization kunnen we dan klikken op “Create Virtual Machines” (eventueel na het installeren van de libvirt modules via `yast -i libvirt`). Er start nu een assistent waar we kunnen kiezen om een nieuw besturingssysteem te installeren. De voorgestelde samenvatting is een visualisatie van het configuratiebestand en kan aangepast worden door te klikken op de titel. Zo passen we bijvoorbeeld de installatiebron voor het OS aan. Na het klikken op Finish start een nieuw console venster waar de grafische installatie dan start.



Figuur 62: SLES10 Service Pack1 installatie onder Xen (via YaST); let op de ‘x’ voor de apparaten

Na de installatie krijgt de machine een IP waarlangs we kunnen verbinden en ze volledig opzetten. Let er wel op om het configuratiebestand (zoals gezegd te vinden onder `/etc/xen/vm/<domain-name>`) aan te passen zodat de cd niet telkens gemount wordt als het systeem start.

5.3. Xen eigenschappen en optimalisaties

5.3.1. Xen (domU) systeem dupliceren

Voor sommige gevallen is het handig als we een bestaand systeem perfect willen dupliceren (of klonen) zodat we met exacte kopieën kunnen werken. Dit is echter niet mogelijk via YaST zodat we het via de console zullen moeten doen.

- ◆ We kopiëren eerst de virtuele schijf, dit lukt het best met een file-image, met een van de fysieke methoden zouden we een kopie moeten maken met de *dd*-tool.
 - `cd /var/lib/xen/images`
 - `mkdir duplicate-sles10`
 - `cp xen-sles10/disk0 duplicate-sles10/disk0`
- ◆ Daarna kunnen we het gemakkelijkst YaST gebruiken om een nieuwe configuratie aan te maken, zoniet kopiëren we het vorige configuratie-bestand en passen we de name, uuid en vif aan. Uuid kunnen we weglaten en voor vif kunnen we `vif = ['']` invullen, deze worden dan automatisch aangemaakt bij de eerste start.
- ◆ Tenslotte kunnen we het gekopieerde domain al starten, maar het netwerk zal echter nog niet meteen werken, het vorige mac-adres zit namelijk nog in het interne systeem. Dit dienen we in te stellen in `/etc/sysconfig/network/<interface>` zodat alles correct overeenkomt. We kunnen tevens de hostname aanpassen (via YaST). Nu nog een laatste herstart en het domain is gedupliceerd.

5.3.2. Voor- en nadelen

5.3.2.1. Voordelen

De prijs: Xen is gratis en als de Linux-ervaring er is, dan is het zeker het proberen waard. Xen is even stabiel als Linux en heeft geen last van virussen of dergelijke.

Transparantie: Xen is opensource en dus mag iedereen eraan meewerken of prutsen. Net zoals Linux is ook de werking van Xen volledig bestands gebaseerd en dus kunnen ook alle wijzigingen met een simpele tekstuele editor aangepast worden.

Scripting: alles kan via de command line uitgevoerd worden en dus is het relatief simpel om scripts te schrijven. Zo wordt bijvoorbeeld backuppen eenvoudig en kosteloos.

5.3.2.2. Nadelen

Xen heeft veel tekortkomingen en gaat nog zeer dikwijls gepaard met onverklaarbare problemen. Het vereist een Linux-kenner om alle problemen te overkomen, want vele optie's zijn niet mogelijk via de grafisch beschikbare tools. Dit maakt de instapdrempel voor Xen heel hoog. Bovendien is Xen nog in een snel tempo aan het ontwikkelen, wat betekent dat de documentatie geregeld achterloopt.

Let erop dat Xen 2 en Xen 3 niet onderling compatibel zijn, dit betekent dat images die werken onder Xen 2 opnieuw zullen moeten worden geïnstalleerd onder Xen 3.

32-bit systemen en 64-bit systemen kunnen nog maar sinds versie 3.0.2 naast elkaar draaien in geparavirtualiseerde mode. 64-bit ondersteuning is er officieel wel, maar in de praktijk loopt dit soms ietwat stroef.

5.3.3. Virtuele schijven op aparte schijf

Net zoals bij andere hardware platformen is het een best practice om de harde schijfbestanden te plaatsen op een apart schijfsysteem. Dit is zeer eenvoudig doordat het gewoon een kwestie is van de bestanden kopiëren en dan die ene regel in het configuratie-bestand aan te passen. Ook het toevoegen van een extra schijf (om bijvoorbeeld database data apart te zetten binnen

een domain) is gemakkelijk uit te voeren. We kunnen bijv. de fysieke raid-array meteen als schijf met het domain verbinden (*phy:/dev/sda1,sda2,w*).

5.3.4. Logical Volume Manager

Er is nog een best practice die wellicht ook voor prestatiewinst zorgt. Bij de meeste Linux distributies is er standaard ondersteuning voor logische volumes. Logische volumes zijn tot op zekere hoogte vergelijkbaar met software RAID, het ondersteunt zelfs de striping & mirroring functies die respectievelijk RAID0 & RAID1 bieden. Het kan echter nog meer, het kan werken met snapshots, read-only volumes en het is zeer flexibel om te veranderen qua grootte of zelfs splitsen & samenvoegen. Enkel pariteits-controle is niet mogelijk.

Uiteindelijk kan LVM werken op zowel MBR als GPT-partities. Het maakt een speciale LVM header aan, in het begin van de partitie en zal er dan voor zorgen dat de volumes altijd starten op een sector die een veelvoud van 64Kbyte is. Op die manier is er een auto alignment en dienen we ook hier niet meer op te letten tenzij we een speciale offset willen. In het laatste geval kunnen we een veelvoud van 64Kbyte meegeven met de *--metadatasize <size>* parameter. Als we dan bijvoorbeeld 230 meegeven zal hij starten op 256 (= veelvoud van 64). In die header staan alle gegevens plus alle unieke IDs (UUID) van de volumes.

6. Microsoft Hyper-V, last but not least

Ook Microsoft doet (misschien te laat) een poging om zich te handhaven op de nieuwe markt der virtualisatie. Geen beter platform om dit anno 2008 op te lanceren dan de Windows Server 2008⁵. Hyper-V was eerder aangekondigd onder de naam “Viridian” en is onderdeel van het besturingssysteem Windows Server 2008 dat eigenlijk meer een platform geworden is.

Echter, in volledige Microsoft stijl, niet alle beloofde onderdelen zitten in Windows Server 2008., sommigen zijn er volledig uitgelaten (zoals het WINFS bestandssysteem), anderen zitten erin maar dan als Beta of Release Candidate. Bij die laatste soort hoort Hyper-V, die in de finale versie van Windows Server 2008 inbegrepen werd in Beta stadium..

6.1. Installatie

Uiteraard is de eerste stap, de installatie van Windows Server 2008. Net zoals bij Windows Vista zijn er ook meerdere versies van Windows Server 2008.

Ten eerste is er de keuze tussen 32- en 64bit. Per keuze zijn er dezelfde opties. De tweede keuze is de Core Installatie of de Full Installatie (uitleg in Appendix E). Ook hier zijn er telkens 3 opties. Standard, Enterprise & Datacenter edition. Ze zijn alledrie uitgerust met de Hyper-V technologie, maar de duurdere versies (Enterprise & Datacenter) hebben extra opties zoals Cluster-ondersteuning.

6.1.1. Vereisten

Voor Windows Server 2008 in het algemeen zijn volgende systeemvereisten gesteld:

Tabel 2: Systeemeisen voor Windows Server 2008

Onderdeel	Absolute minimum vereisten	Maximum limiet of aangeraden hoeveelheid
Processor	1GHz (x86) of 1.4GHz (x64)	2GHz of hoger is aangeraden
Geheugen	512MB	2GB of meer is nodig, met max tot 64GB voor x86 en 2 TB voor x64 systemen
Harde schijf ruimte	10GB	40GB of meer is aangeraden, afhankelijk van opties en applicaties
Optisch station	DVD-ROM lezer	
Scherf	SVGA (800x600)	

⁵ Zie ook appendix E: Windows Server 2008 Overzicht

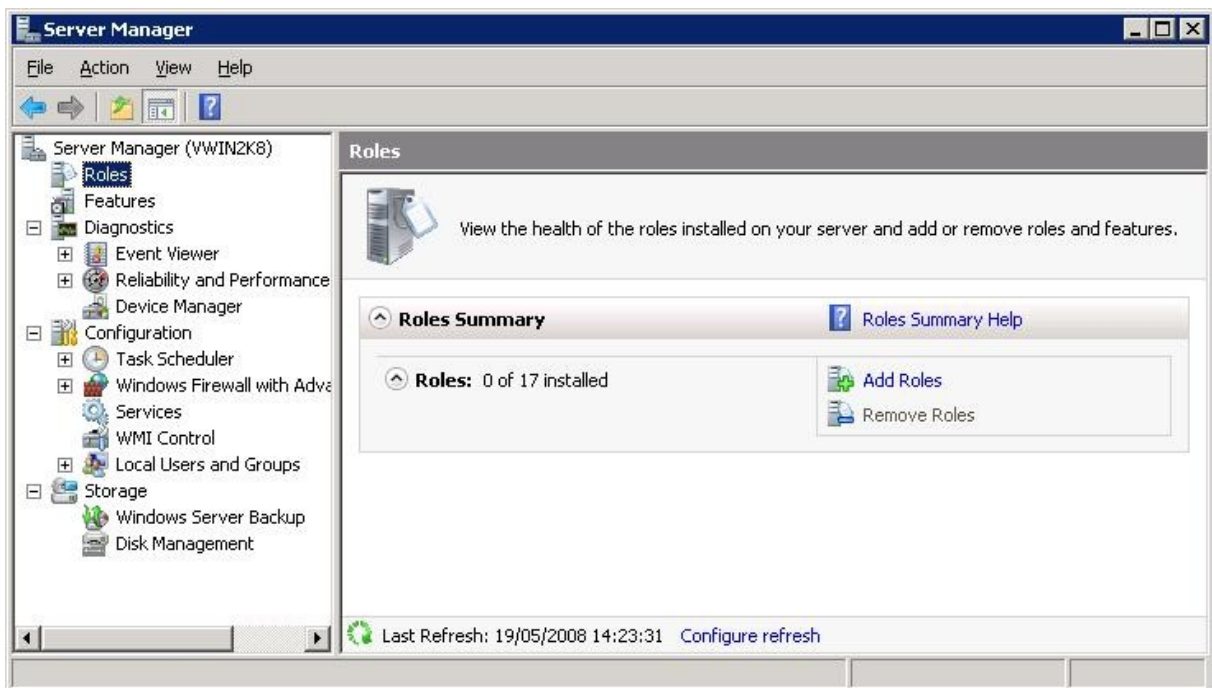
Let echter op, voor Hyper-V zijn andere vereisten van toepassing. Aangezien Hyper-V gebruik maakt van onder andere Hardware Virtualisatie is een Virtualization Enabled CPU nodig. Een tabel met een overzicht van deze CPUs is te vinden in tabel 1.

Let erop dat dit tevens moet *enabled* staan binnen de BIOS van de server waarop Hyper-V moet komen. Deze CPUs zijn allemaal 64-bit CPUs en dus is een vereiste voor Hyper-V ook een 64bit versie van Windows Server 2008.

6.1.2. Concreet

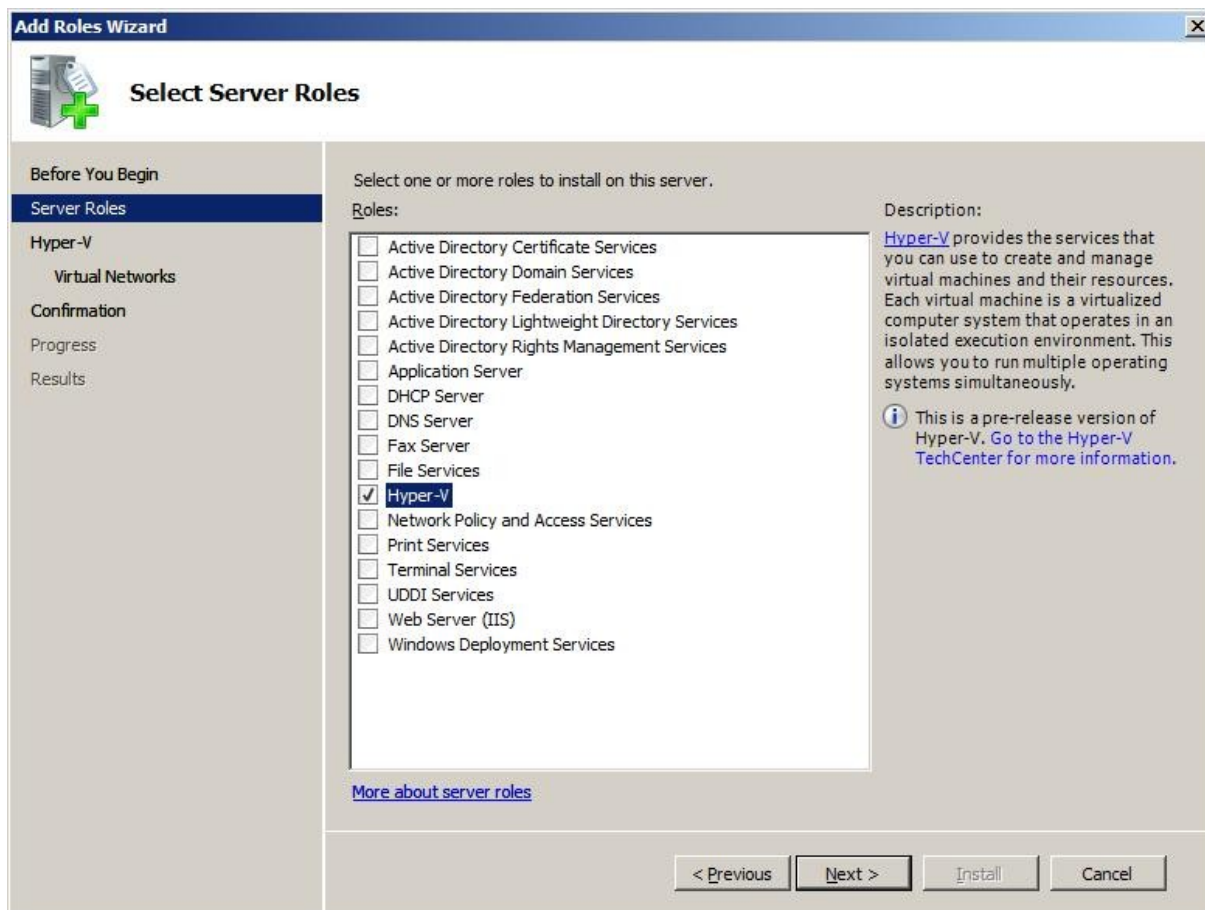
Na het installeren van Windows Server 2008 (zie ook Appendix E) kunnen we een rol toevoegen, namelijk die van Hyper-V. Let erop dat Hyper-V meteen al enkele controles doet, zoals de CPU-ondersteuning. Als dit niet klopt zal Hyper-V weigeren te installeren.

Zoals de meeste producten van Windows, of Microsoft in het algemeen, doen we dit aan de hand van een wizard.



Figuur 63: Windows Server 2008: Server manager

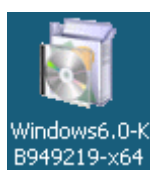
Na het klikken op “Add Roles” kunnen we uit de 17 mogelijke “rollen” die dit besturingssysteem kan spelen. We kiezen uiteraard voor de Hyper-V en hopen dat alle checks overleefd worden. Bij het volgende venster kunnen we kiezen of we een Virtueel Netwerk willen maken en zoja, met welke fysieke netwerkkaart we het willen verbinden. Dit is als een Virtuele Switch die verbonden is met de eventuele fysieke netwerkkaart van de server. Dan nog bevestigen en na een herstart is Beta Hyper-V geïnstalleerd.



Figuur 64: Windows Server 2008: Voeg rol "Hyper-V" toe

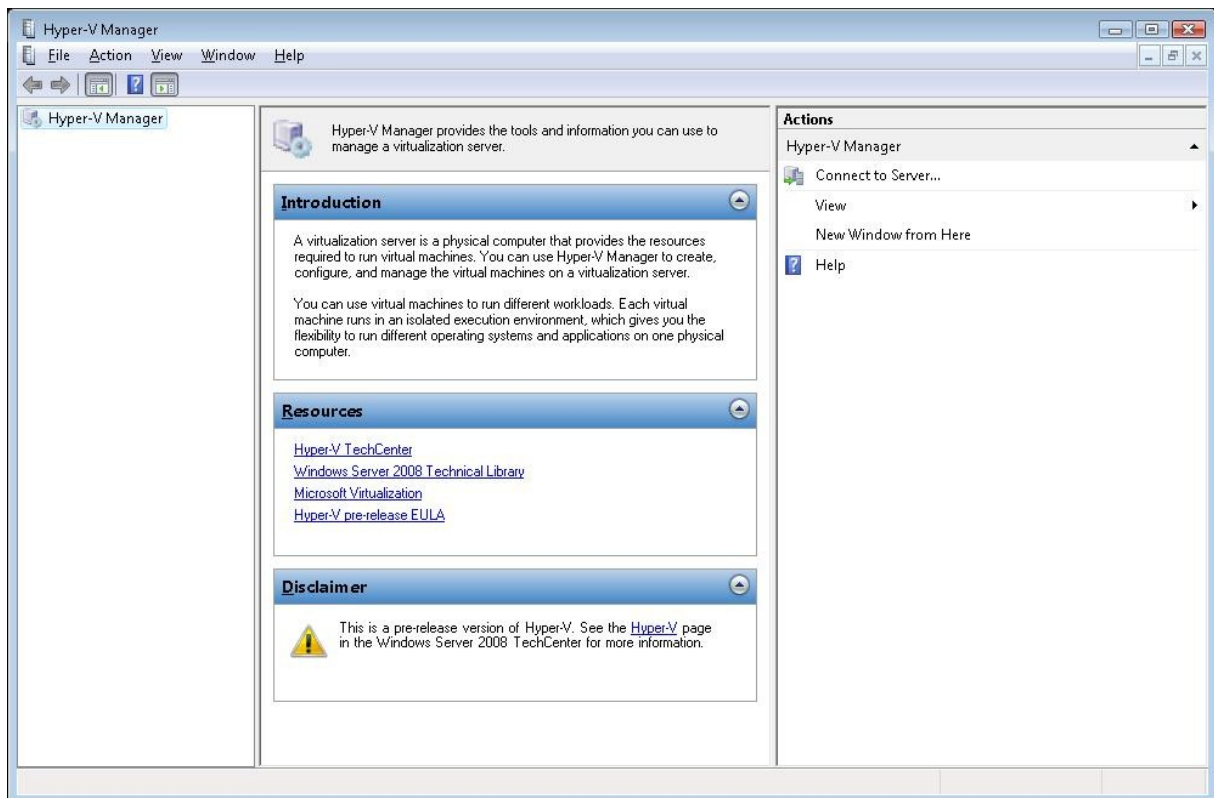
Na de installatie merken we dat de Hyper-V service niet wil starten, dit komt omdat de Beta enkel kan starten als hij ziet dat de server gelokaliseerd is in de United States. Dit kunnen we aanpassen in "Regional & Language Options" en dan zal de service wel starten.

Een 2^{de} en betere oplossing is de update naar de Release Candidate versie. Deze is gratis af te halen van de Microsoft site (KB949219) en vereist tevens een herstart na installatie.



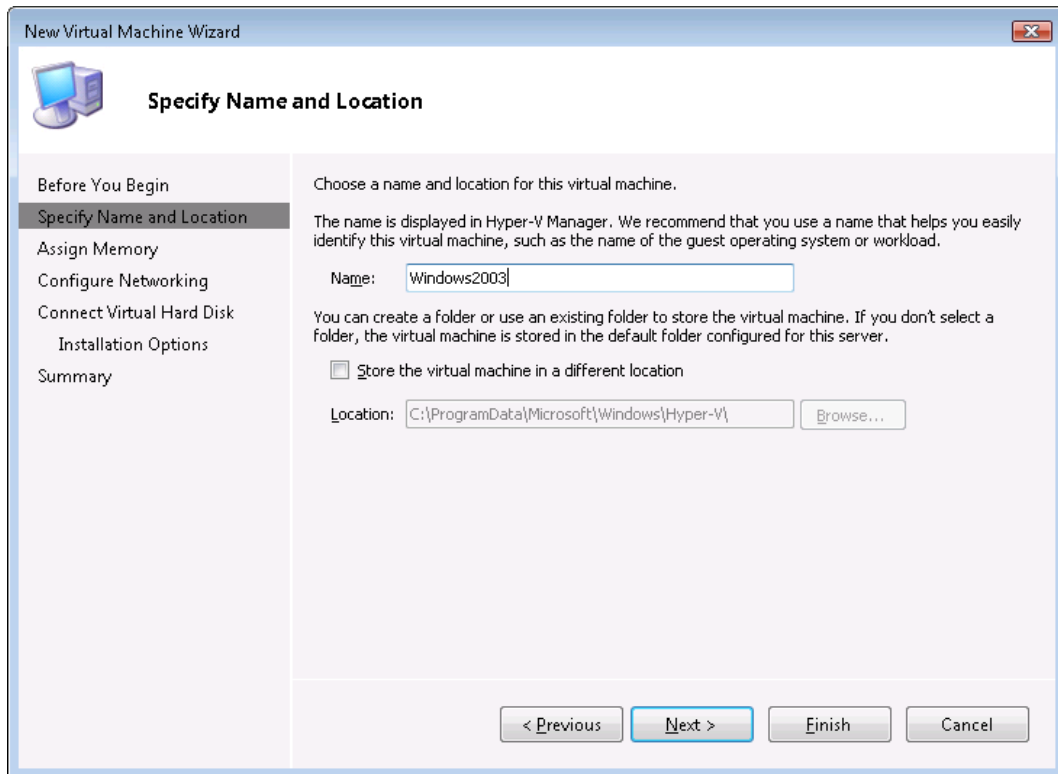
Figuur 65: Windows Server 2008:Hyper-V update naar Release Candidate 0

Na installatie kunnen we Hyper-V volledig beheren via dezelfde tool als waarlangs hij werd geïnstalleerd: de Server Manager. Zoals bijna alles binnen Windows Server 2008 te beheren is via hetzelfde platform: Microsoft Management Console. Wat Hyper-V betreft is er zelfs een MMC-plugin voor andere platformen (zoals Windows Vista SP1) om Hyper-V te beheren vanuit een andere pc in het netwerk.



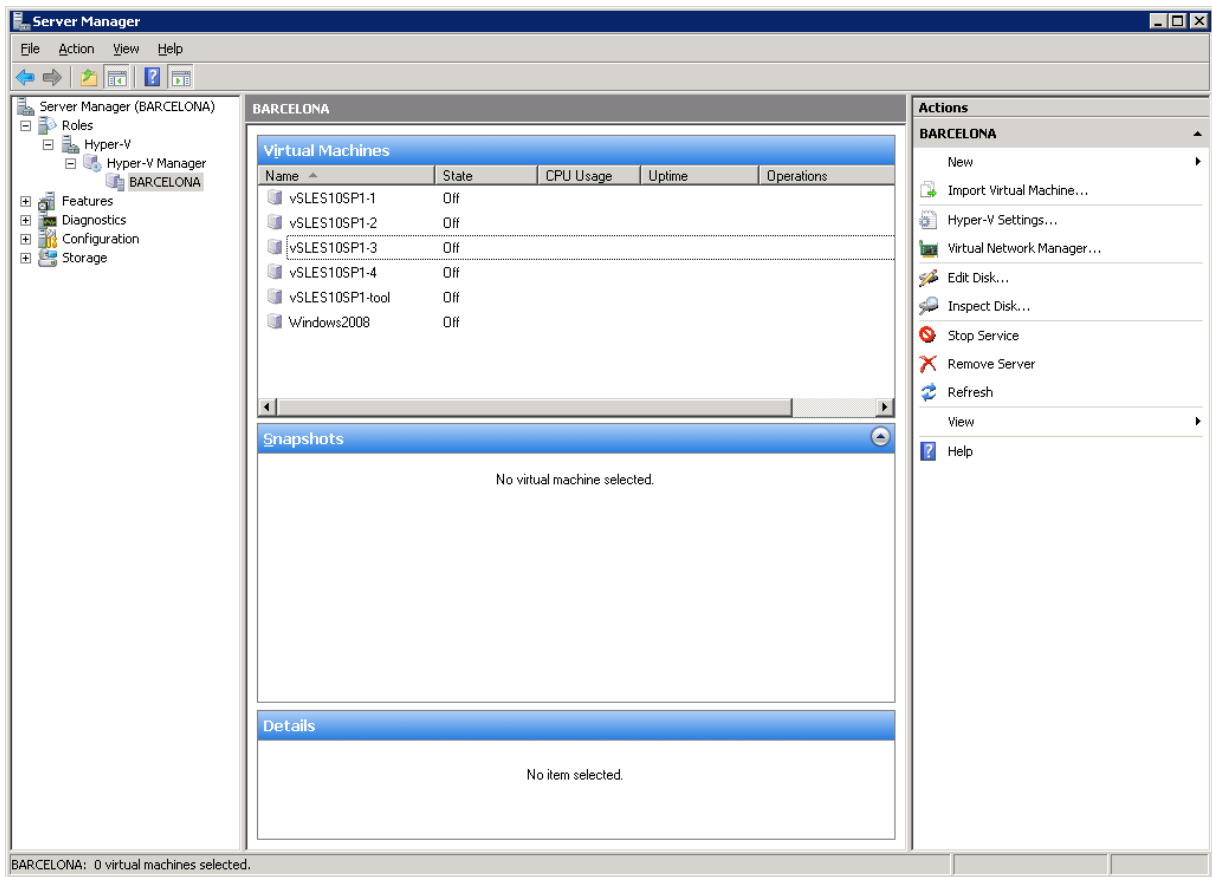
Figuur 66: Hyper-V Manager op Vista

We maken een verbinding met de localhost of een andere host en kunnen dan een nieuwe virtuele machine, harde schijf of diskette schijf aanmaken. Bij het aanmaken van een virtuele machine dienen we zijn naam op te geven en hoeveel geheugen we hem toekennen. We geven ook mee welke schijf we toekennen. In deze wizard kunnen we enkel een nieuwe schijf aanmaken of een bestaande toekennen. Geavanceerde opties als de keuze van controller of fysieke schijf dienen we achteraf te doen.



Figuur 67: Hyper-V: het aanmaken van een virtuele machine

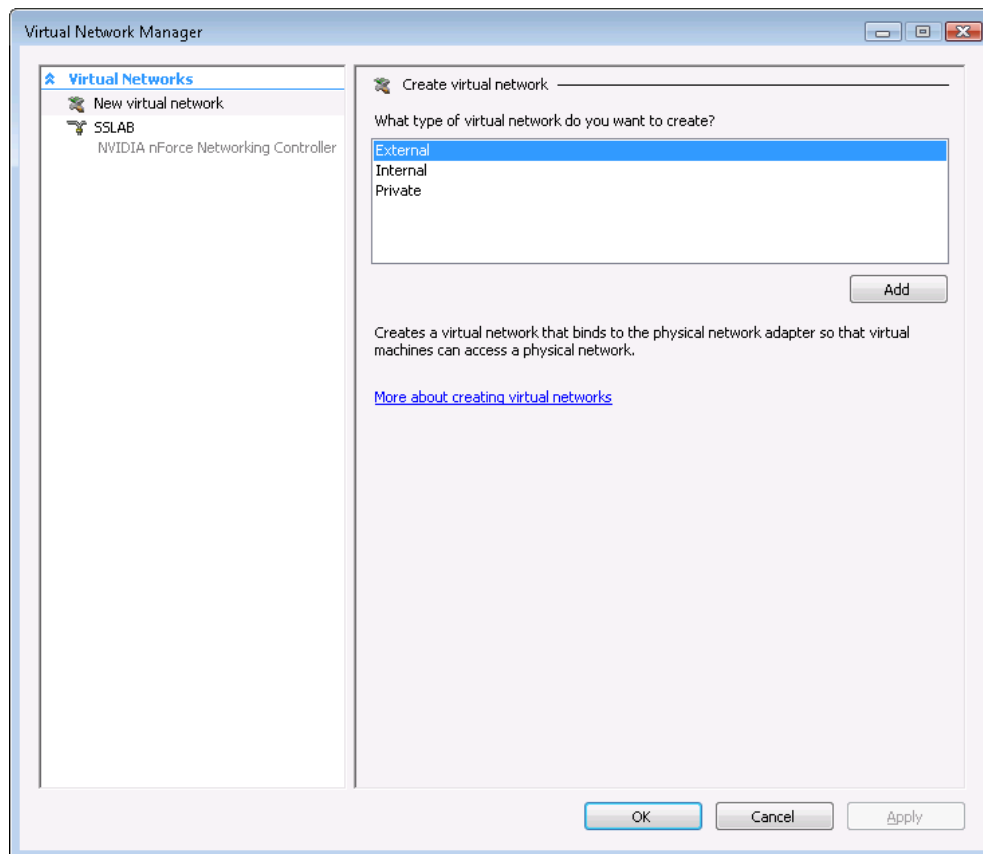
Om een VM een netwerk te kunnen toekennen dienen we eerst zo'n virtueel netwerk aan te maken. Dit kan in het hoofdvenster van de Hyper-V manager (na verbinding met een server).



Figuur 68: Hyper-V: alle opties op een rijtje

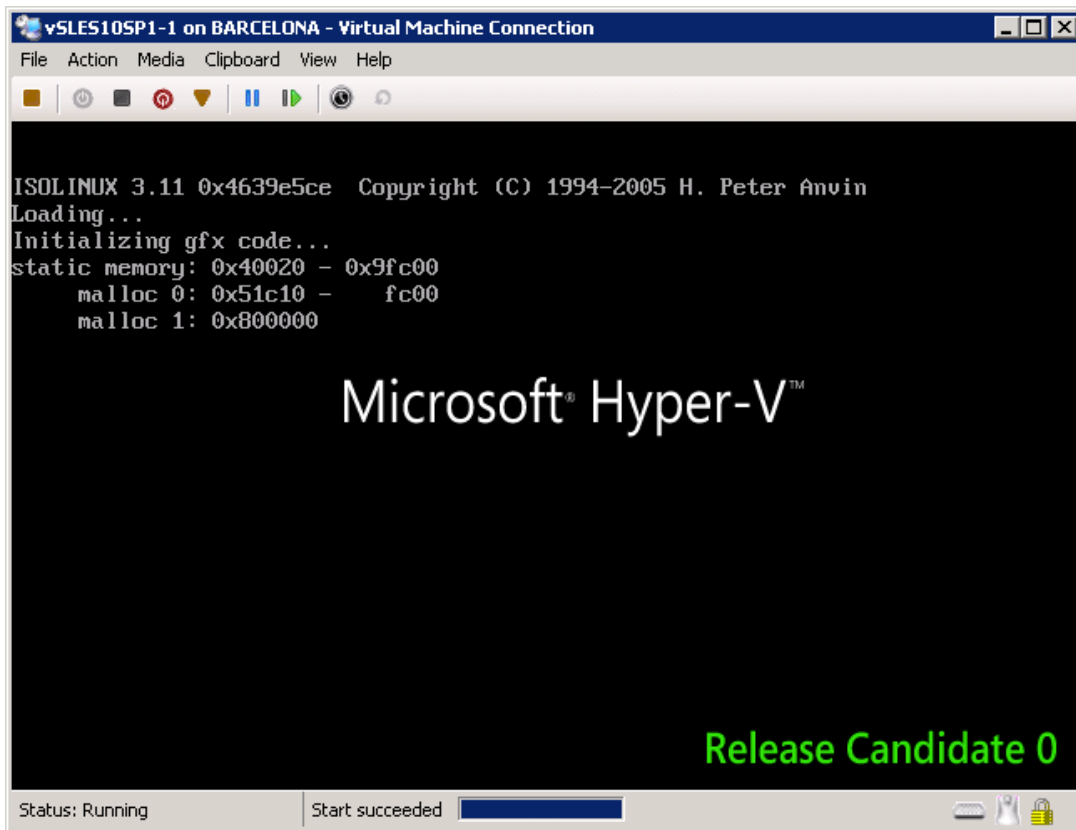
Als we klikken op “Virtual Network Manager...” kunnen we bestaande netwerken beheren of nieuwe netwerken aanmaken. Er zijn 3 soorten netwerken: interne, externe & private netwerken.

Een extern netwerk wordt verbonden met een fysieke netwerkadapter en kan op die manier met het externe netwerk verbonden worden. Een intern netwerk zorgt ervoor dat de virtuele machines die ermee verbonden zijn, kunnen communiceren met elkaar en met de fysieke host. Een privaat netwerk voorziet enkel in communicatie tussen de virtuele machines die ermee verbonden zijn. En dus niet met de fysieke host en al helemaal niet met een fysiek netwerk.



Figuur 69: Hyper-V: de Virtual Network Manager

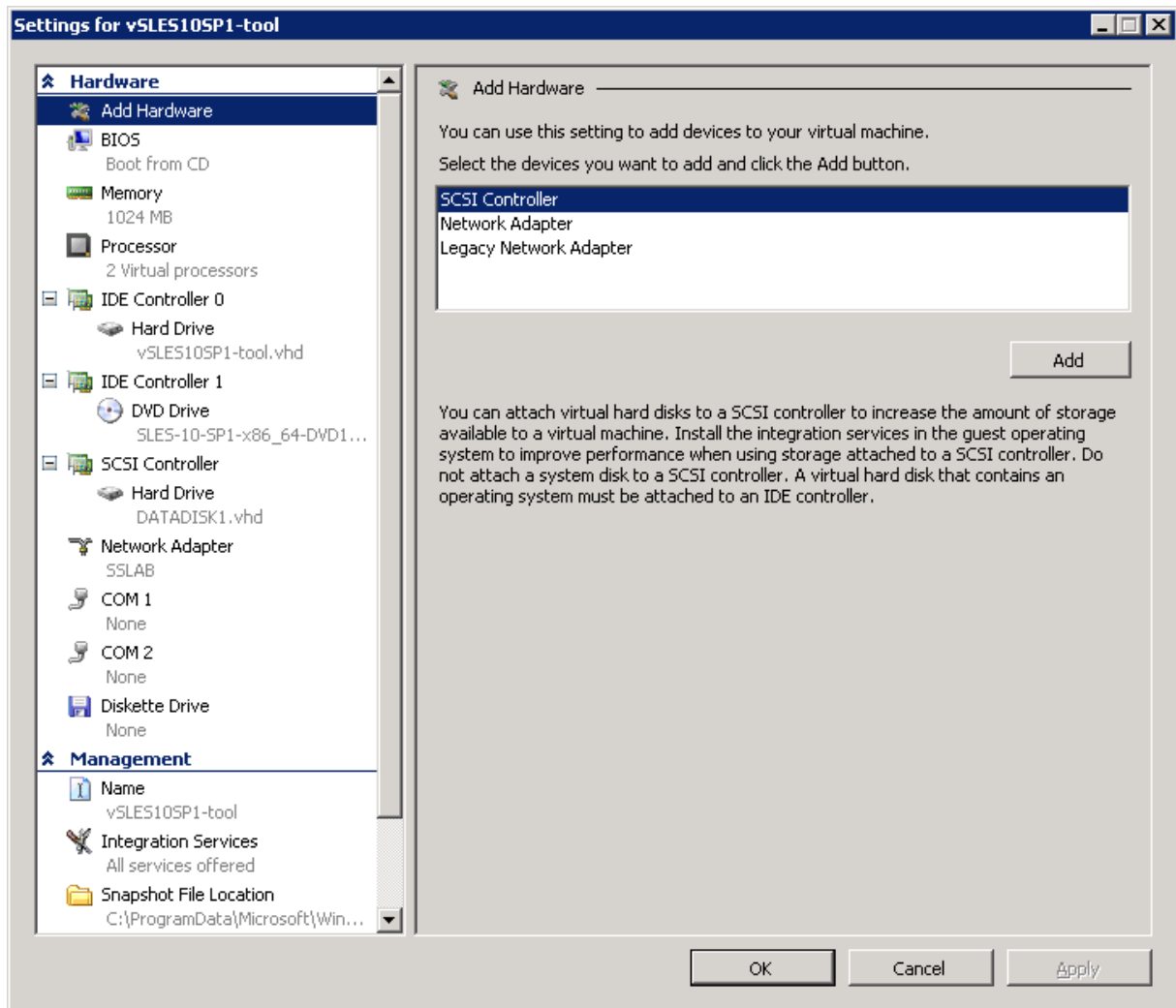
Na het uitvoeren van deze instellingen kunnen we een console starten met de aangemaakte virtuele machine. Via het Action menu kunnen we de machine starten en via het Media menu kunnen we een (installatie-) cd verbinden met de VM.



Figuur 70: Hyper-V: Virtual Machine Console

6.2. Eigenschappen en optimalisatie

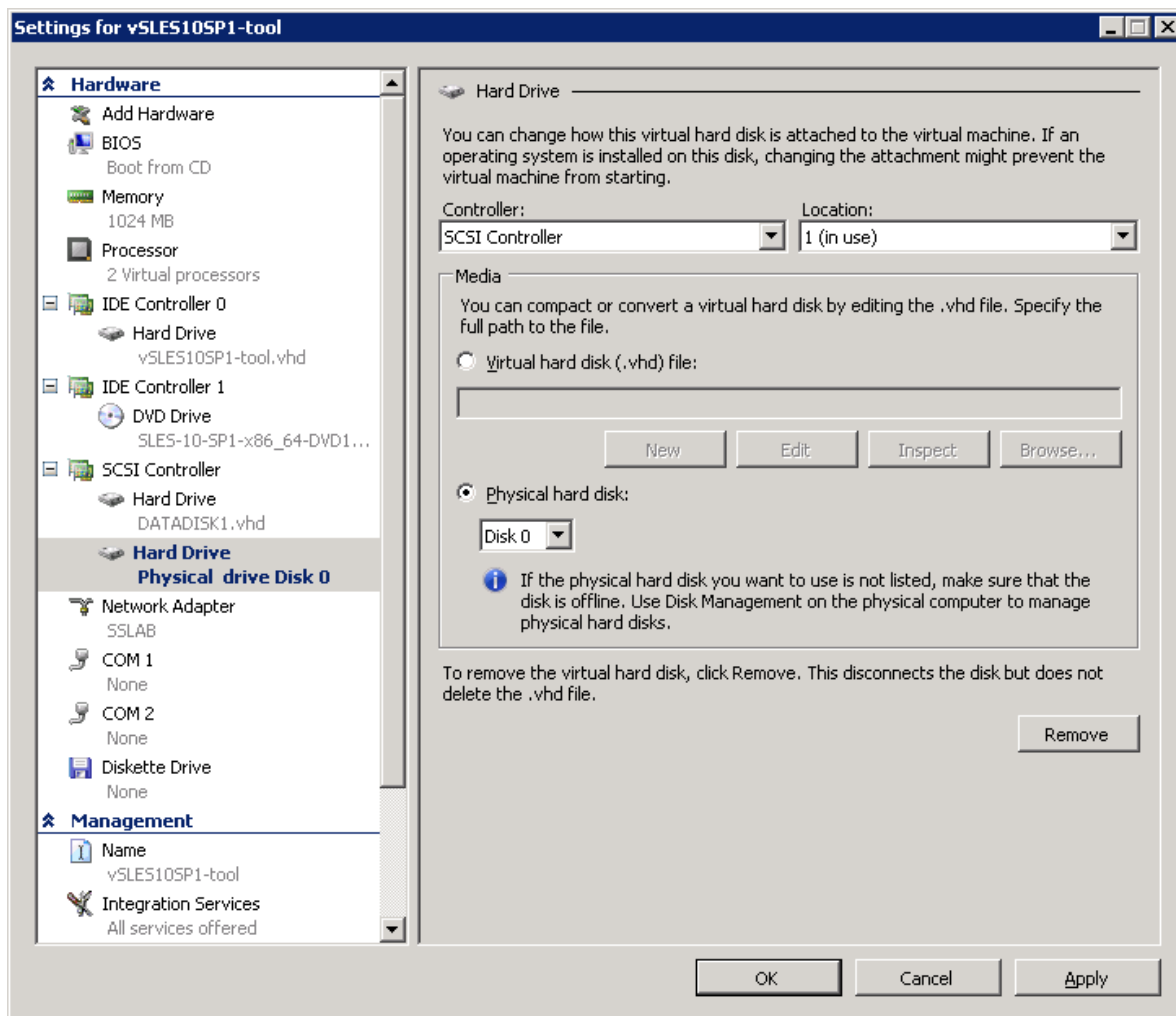
Bij de eigenschappen van een virtuele machine kunnen we veel meer instellen dan via de initialisatie wizard. SCSI Controllers & Network Adapters zullen door SUSE Linux standaard niet worden herkend. Maar straks wordt daarvoor een workaround getoond. Een Legacy Network adapter werkt daarentegen wel meteen. Hier kunnen we ook het aantal verbonden processors kiezen. Een groot nadeel is het feit dat we geen CPU affiniteit kunnen instellen per VM.



Figuur 71: Hyper-V: Virtual Machine Settings

6.2.1. Fysieke schijf gebruiken

Als we bij een bestaande IDE Controller (of een nieuwe SCSI Controller) een schijf willen toevoegen hebben we de keuze tussen een schijf-bestand of een volledige fysieke schijf. We kunnen enkel fysieke schijven selecteren die wel gedetecteerd werden door Windows maar geen stationsletter hebben en op “Offline” staan. Op die manier is Hyper-V zeker dat het exclusieve toegang tot de schijf heeft en er ook geen “vervuiling” kan zijn van het Host OS.



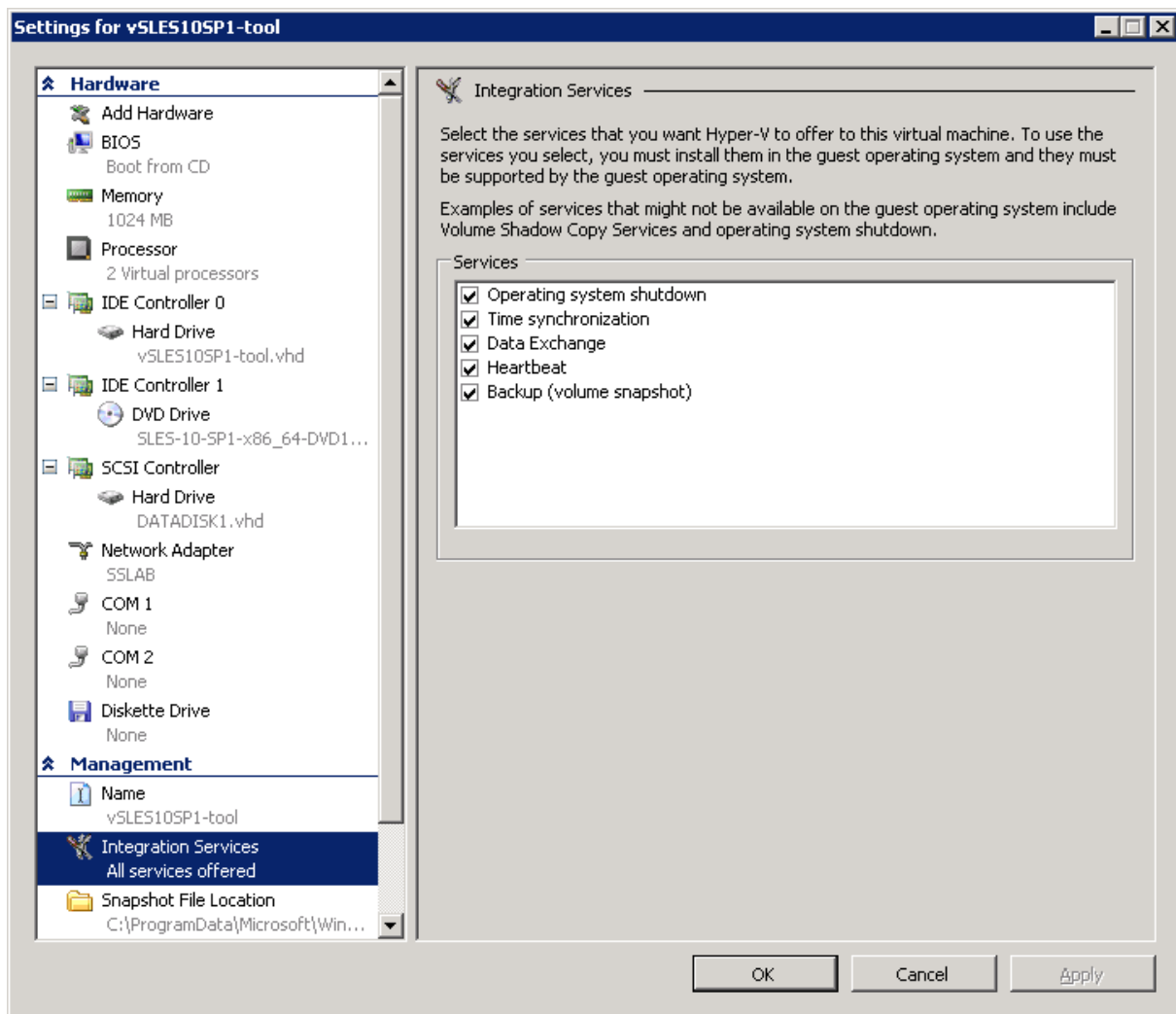
Figuur 72: Hyper-V: een fysieke schijf toevoegen

6.2.2. Integration Services

Net als ESX heeft ook Hyper-V een set tools ontwikkeld om gevirtualiseerde gast systemen te ondersteunen. Ze leveren verbeterde drivers voor geëmuleerde apparaten, tijdssynchronisatie, data uitwisseling en nog meer...

Deze aparte opties kunnen aan- of uitgevinkt worden na installatie in de Virtual Machine Settings. Let er wel op dat deze dingen énkél van toepassing zijn op Windows Gast Systemen. Microsoft biedt ondersteuning voor Windows Server 2003 SP2, Windows Server 2008, Windows XP SP3 & Windows Vista SP1.

Als we een console starten kunnen we deze componenten installeren via Media > Insert Integration Services Setup Disk. Binnen het gastsysteem is het dan enkel een kwestie van de setup.exe te starten en te installeren.

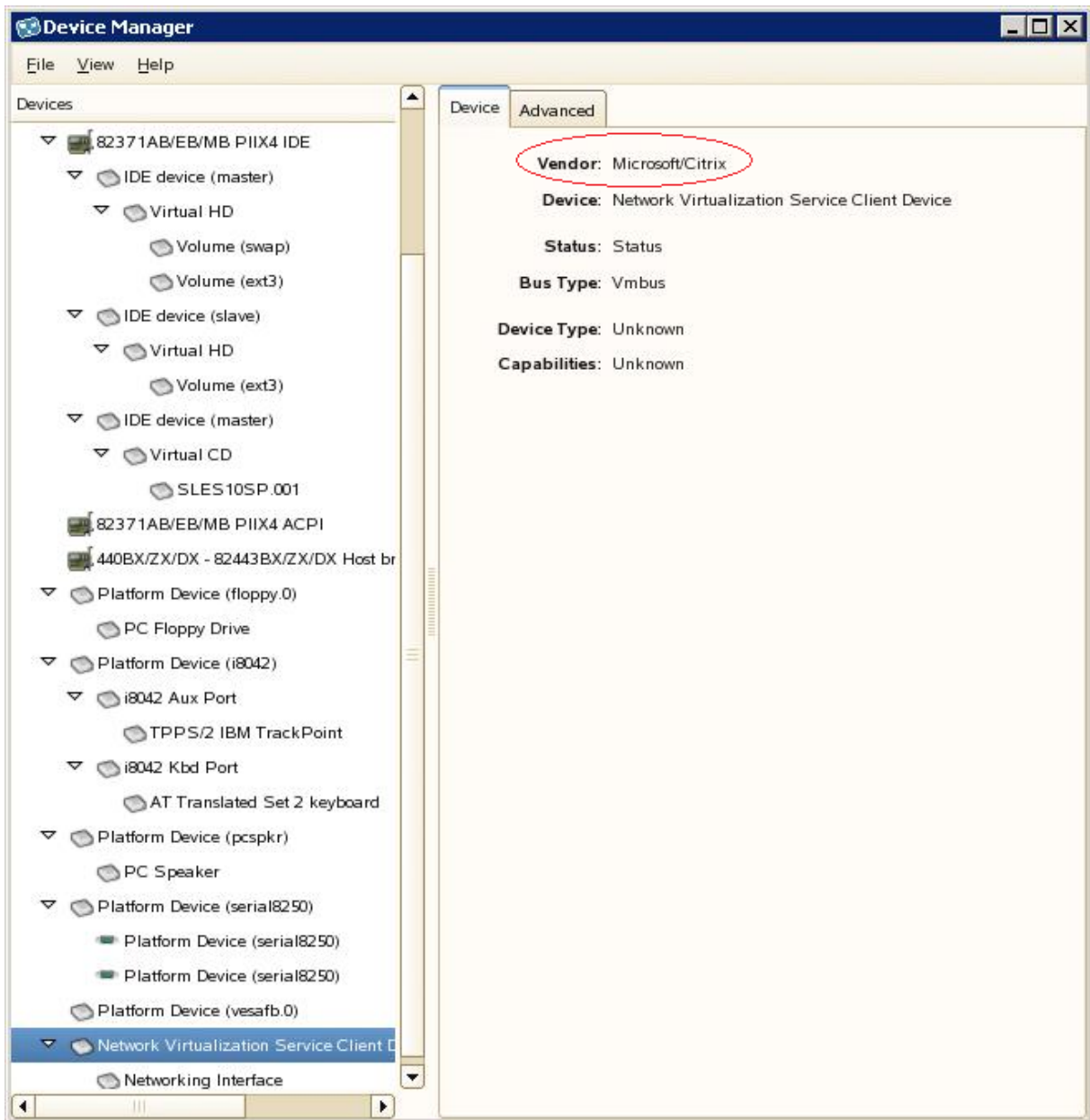


Figuur 73: Hyper-V Integration Services

Er is ook een ondersteuning voor Novell SUSE Linux Enterprise Server 10 Service Pack 1 32bit én 64bit. Deze Integration Tools zitten niet standaard in Hyper-V maar zijn wel van de Microsoft website te downloaden.

De manier waarop Hyper-V dit aanpakt is door gebruik te maken van de Xen techniek (die bezit is van Citrix). Zowel hun SCSI Controller als hun Network Controller (niet de Legacy controller) worden dus geparavirtualiseerd. Hiervoor is een behoorlijk ingewikkeld stappenplan nodig, want het omvat het hercompileren van een xen-kernel.

Opgelet! Werkt enkel op de SLES met kernel 2.6.46-0.12 en niet op de geupdate versie.



Figuur 74: Hyper-V: Apparaatbeheer van geparavirtualiseerde SLES10

7. Benchmarking Virtualisatie: theorie in de praktijk met resultaten

Het lab waarbinnen de masterproef plaatsvond profileert zich als een serversizing, server stresstesting lab. Dit betekent dat het zich specialiseert in het onderzoeken en optimaliseren van servers door ze daadwerkelijk te testen en hun workloads zo hoog mogelijk te brengen in diverse omstandigheden.

Er zijn veel soorten benchmarks: schijfsnelheden, CPU prestaties, geheugen, videokaarten...

Voor alles is er wel een aparte benchmark. Maar dit is niet het doel van de masterproef. We willen een real-life omgeving simuleren en KMO's hebben geen baat met de wetenschap dat de ene server betere geheugen-prestaties heeft dan de andere.

Een veel gebruikt type benchmarks zijn de zogenaamde OLTP-benchmarks, wat staat voor een Online Transaction Protocol benchmark. Deze doet typisch een stresstest van een bepaalde database en simuleert gebruikers (websites/applicaties) die queries uitvoeren op deze database. Hij kijkt daarbij onder andere naar de antwoordtijd (responsetime) of doorvoersnelheid (throughput).

Om testen te kunnen uitvoeren is er uiteraard hardware nodig. Deze speciale hardware⁶ (servers met storage adapters) wordt uitgebreid besproken in Appendix F: Hardware overzicht. Qua servers is het enige onderscheid de CPUs dus zullen we eerst deze hier benoemen en met deze codenamen verder werken.

- *Codename Clovertown: Dual Quadcore Xeon E5345 @ 2,33GHz*
- *Codename Harpertown : Dual Quadcore Xeon E5472 @ 3GHz*
- *Codename Barcelona: Dual Quadcore Opteron 2350 @ 2GHz*

De E5472 is een nieuwere generatie Xeons dan de E5345 en moet ons de vooruitgang tonen die Intel geboekt heeft. Het grootste verschil is het feit dat de Harpertown gebouwd is op het 45nm-proces terwijl de Clovertown nog een 65nm-proces gebruikte. Maar er is veel meer verschil tussen de 2 dan dat. Hij heeft bijvoorbeeld een 4-bit Radix-16 divider in plaats van een 2-bit versie bij de vorige generatie. Verdere verschillen betreffen: 2 x 6MB L2 Cache i.p.v. 2x4MB, nieuwere SSE-versie en 1600MHz FSB ipv 1333MHz.

De Barcelona is de eerste editie van AMDs Quad Core CPUs en moet de strijd aangaan met de Quad Cores van Intel. Enkele grote voordelen van deze CPU is het feit dat hij de frequentie van iedere kern apart kan instellen naargelang de belasting. Hij kan SSE128-instructies in 1 klokcyclus uitvoeren en iedere kern heeft een L2 cache van 512KB plus een gedeelde L3 cache van 2MB. Er is echter een probleem met deze Opteron 2350-versie, er zit een vervelende TLB-Bug in. Deze zorgt voor een deadlock (betekend dat alle threads op elkaar zitten te wachten) in het geval van recursief of genest schrijven naar de cache. Zoals we straks zullen zien betekent dit dat we de AMD resultaten niet kunnen vergelijken met de Intel

⁶ Appendix D: Hardware overzicht

CPUs. De nieuwe B3-stepping van de 2350 zou deze problemen oplossen, maar was op tijd van schrijven nog niet beschikbaar.

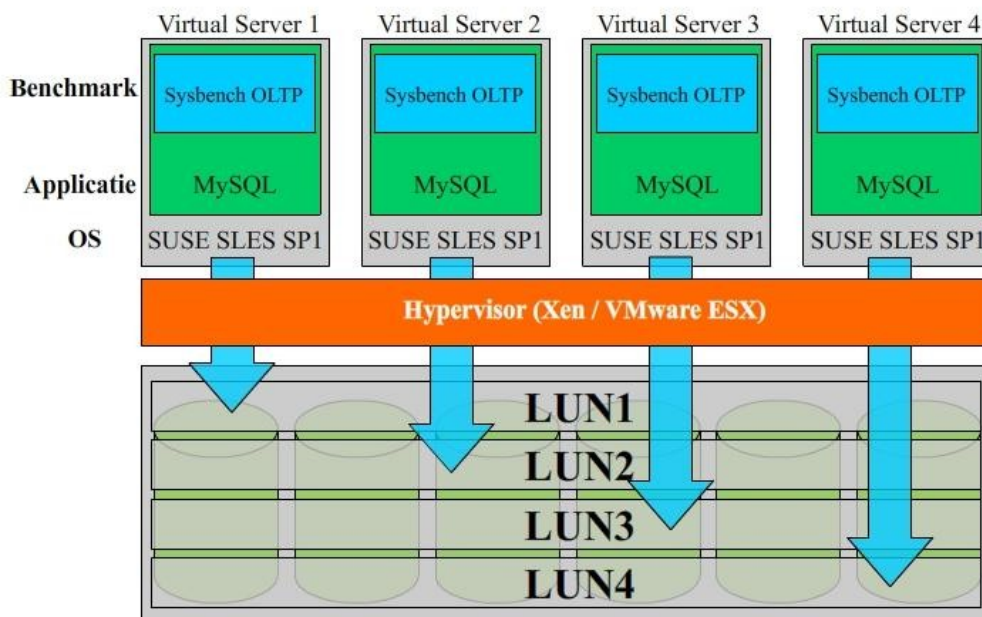
7.1. Opzet, testfactoren

Voor de eerste serie testen werd één tool gebruikt, namelijk Sysbench. Deze OLTP-benchmark zal een MySQL tabel aanmaken en populaten (vullen) met een opgegeven aantal records en hierop dan lees- & schrijf-opdrachten (SELECT, INSERT, UPDATE & DELETE) uitvoeren. Het is een multi-threaded tool wat hem uitstekend maakt voor virtualisatie-testing. Zo wordt zowel CPU als harde schijf getest, maar we zijn vooral geïnteresseerd in hoe de CPU presteert. Daarom gaan we de harde schijf zo snel mogelijk maken door de database te zetten op een DAS-systeem. Dit betreft een 8-schijven RAID0 systeem. Op die manier gaan we de CPUs als beperkende factor nemen.

Onze test-opzet is als volgt: we testen per machine eerst Sysbench native, d.w.z. we installeren Novell SUSE Linux Enterprise Server 10 Service Pack 1 (SLES) en laten Sysbench werken met de MySQL databases op het RAID systeem.

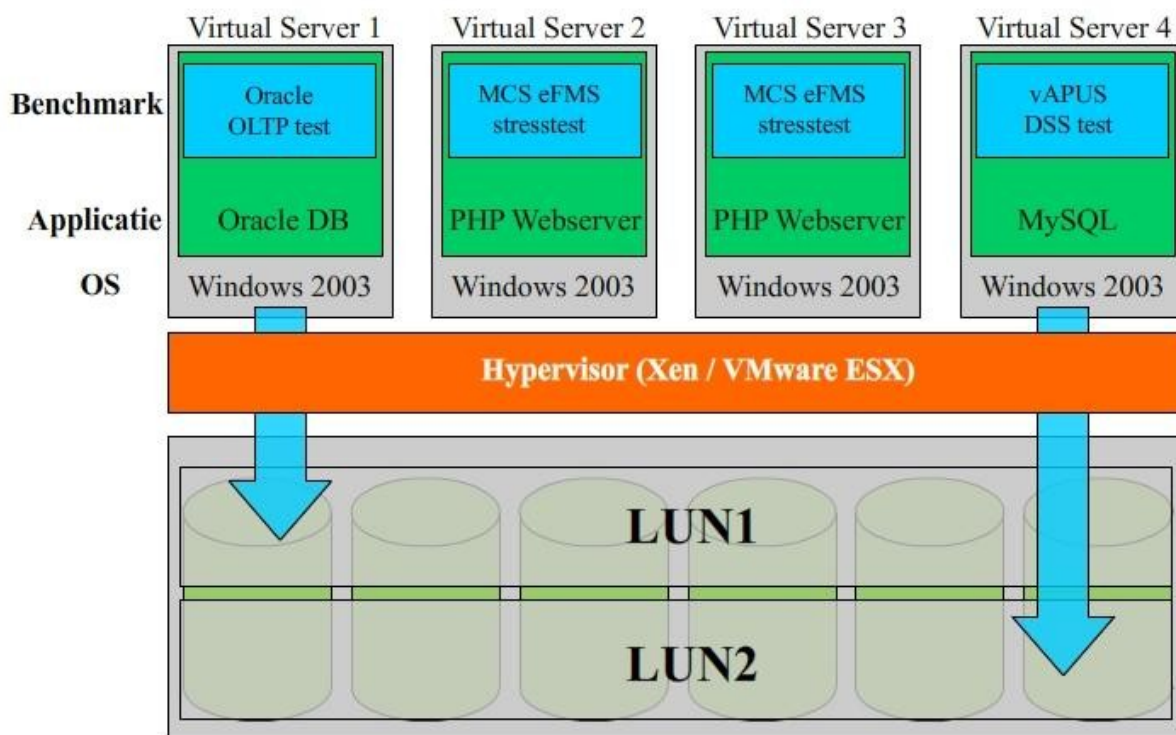
Na de native test installeren we het virtualisatie-platform en zetten 1 virtuele SLES-machin op en laten Sysbench ook werken op het RAID systeem. Doordat de CPU - commando's gevirtualiseerd moeten verlopen hopen we op die manier de impact te zien van het virtualisatieplatform.

Tot slot proberen we nog consolidatie te simuleren door 4 van dergelijke virtuele machines naast elkaar te draaien en de Sysbench test te herhalen op de 4 machines tegelijk. Alle 4 draaien op het RAID systeem. Dit staat ook wel bekend als het Worst-Case scenario omdat een bedrijf nooit 4 loodzware database toepassingen op eenzelfde fysieke machine zal plaatsen.



Figuur 75: Worst Case Test-scenario

Er werd ook gebruik gemaakt van een 2^{de} test scenario, dit om de realistische wereld meer te benaderen. Dit houdt in dat verschillende soorten applicaties naast elkaar en tegelijk worden getest. Het gebruikte OS op alle virtuele machine is Windows Server 2003 Enterprise. Er zijn onder andere twee database intensieve applicaties, een OLTP en een DSS (Decision Support System). Dit DSS, op basis van OLAP (Online Analytical Processing), is een computer gebaseerd systeem om mensen aan de hand van verschillende technologieën beslissingen te helpen nemen. OLAP is een methode om snel een antwoord te kunnen bieden op complexe vragen door een veelheid van gegevens in een database te verwerken. De DSS-test omvat een MySQL benchmark ontwikkeld door het lab zelf, codenaam Heimdall en onderdeel van het vAPUS testplatform. Het simuleert gelijktijdige gebruikers die de MySQL database ondervragen. Voor de OLTP test werd de gratis tool Swingbench gebruikt, en dan meerbepaald zijn comand line gebaseerd onderdeel Charbench, samen met een Oracle database. Daarnaast zijn er ook twee virtuele webserver, tweemaal IIS (Internet Information Services), PHP en een Oracle database back-end. Deze database staat op een 2^{de} server en wordt zelden of nooit intensief gebruikt, het netwerk is zeker geen beperkende factor. De factoren die we meten zijn op die manier CPU & I/O intensieve operaties waarbij de gebruikte virtualisatie-overhead en gebruikte CPU architectuur een rol speelt. De gebruikte tool voor de webtests is het testplatform dat door het lab zelf werd ontwikkeld namelijk vAPUS. Dit tweede Real World testscenario kon slechts op een beperkt aantal platformen en met een beperkt aantal hypervisors uitgevoerd worden wegens tijdsgebrek.



Figuur 76: Real World Tests scenario

7.1.1. Prestatiefactoren voor Sysbench

- CPU-limiet: als we de overhead van de virtualisatie platformen willen vergelijken zullen we ervoor moeten zorgen dat de Sysbench test niet alle 8 CPUs gebruikt die hij tot zijn beschikking heeft. We hebben in totaal telkens 8 kernen ter beschikking en dus testen limiteren we Sysbench op 2 CPUs. De grootste CPU verbruiker is echter niet het Sysbench proces maar uiteraard wel het MySQL proces. Bij gevolg dient de CPU-limiet in het opstartscript van MySQL te staan. Voor concrete uitwerking: zie stappenplan.
- Oltp-table-size: dit is het aantal records in de database, hoe groter dit is hoe langer het duurt om de tabel te vullen maar ook hoe willekeuriger de schijf-toegang is. Daar dit geen schijfstest is kiezen we een waarde die groot genoeg is om een KMO te simuleren, namelijk 1 000 000.
- Threads: we dienen ook het aantal threads mee te geven waarmee we willen testen. Dit komt neer op het aantal gelijktijdige gebruikers (websites/applicaties) die de queries simultaan uitvoeren op de database. Deze laten we variëren van 8 tot 32, met stappen van 8. (dus 8, 16, 24 & 32).
- Requests: we moeten ook meegeven hoeveel requests elke gebruiker (website/applicatie) moet uitvoeren. Uiteindelijk zal dit bepalen hoe lang de test duurt. Liefst niet te hoog, anders duurt de test te lang. Maar ook niet te laag, anders zijn de resultaten niet betrouwbaar. We stelden het in op 25 000.

Verdere parameters die Sysbench nodig heeft:

- mysql-table-engine: er zijn meerdere engines die op MySQL van toepassing zijn, maar voor transacties is InnoDB de beste en bekendste.
- mysql-socket: de locatie van het socket bestand. Standaard is dit */var/lib/mysql/mysql.sock*.
- mysql-user & mysql-password: de gebruiker en zijn wachtwoord om te kunnen verbinden met de database.

Een Sysbench test verloopt standaard in 3 stappen:

1. Prepare stap: door de prepare parameter mee te geven met Sysbench zal hij de test voorbereiden. Sysbench veronderstelt reeds de aanwezigheid van een database met de naam *sptest*. In de prepare stap zal hij hierin een tabel aanmaken en vullen met het meegegeven aantal records.
2. Run stap: bij deze stap moeten alle voorgaande opties meegegeven worden, hier zal hij dus de test zelf uitvoeren. En hij zal dit gewoon tonen op het scherm. Dus standaard niet opslaan.
3. Cleanup stap: als we tenslotte nogmaals Sysbench aanspreken met de cleanup parameter zal hij alle tabellen in de *sptest* database verwijderen.

7.1.2. Prestatiefactoren voor MySQL

MySQL optimalisaties zijn in deze stap zeer beperkt om de eenvoudige reden dat ook de gemiddelde KMO geen kennis in huis heeft om ingewikkelde tuning te doen. Het beperkt zicht tot 2 best-practices:

- MySQL Versie: eerdere onderzoeken (tevens door hetzelfde lab uitgevoerd) hebben uitgewezen dat de huidige versies van MySQL (5.0x) een mutex-bug hebben waardoor MySQL zeer slecht tot niet meer schaalbaar is als het aantal CPUs meer dan 4 bedraagt. Dit houdt in dat MySQL sterke vooruitgang toont tot en met 4 CPUs ten opzichte van 1

CPU, maar als we met meer dan 4 CPUs werken gaan de prestaties van MySQL terug (sterk) achteruit.

Deze (door MySQL) erkende bug word veroorzaakt door mutex-problemen. Mutex staat voor Mutual Exclusion en is het proces dat zich bezig houdt met de verdeling van locks op onderdelen van de database. Als een gebruiker bezig is een tabel aan te passen kan mutex deze tabel blokkeren voor andere gebruikers zodat geen data dubbel komt te staan.

Na enkele contacten met MySQL (via de lead developer: Peter Zaitsev) blijkt dat de bug opgelost is in MySQL versie 5.1.23 en daarom gebruiken wij ook voor onze tests deze MySQL-versie.

- MySQL configuratie-bestand: bij het starten van MySQL leest het in zijn configuratie-bestand hoeveel geheugen het mag innemen en nog vele andere parameters. We kunnen de prestaties van MySQL gevoelig verbeteren in onze machines met 16GB geheugen door niet de `my-normal.cnf` te gebruiken maar wel de `my-large.cnf`. Deze dient gekopieerd te worden van de locatie `/usr/share/mysql/my-large.cnf` naar `/var/lib/mysql/my.cnf`.

7.1.3. Prestatiefactoren voor SUSE Linux Enterprise Server 10 SP1 x64

Ook voor SLES zijn er enkele prestatiefactoren die we nageleefd hebben:

- Kernel-update: na registratie kunnen we SLES volledig updaten en na enkele keren zien we dat er een nieuwere kernel is. De standaard kernel is 2.6.16.45-0.12 en na de update (waarbij enkele andere drivers ook worden geupdate) is deze 2.6.16.54-0.2.5.
- X afsluiten: al zullen deze eerder beperkt zijn, maar ook de X manager (om het grafische gedeelte van SLES weer te geven) neemt zijn resources in. Het is een algemene best-practice om dit uit te schakelen. Wij doen dit simpelweg door telkens op te starten in runlevel 3 (aan te passen in `/etc/inittab`).
- ext3 ipv reiserfs: reiserfs is het standaard bestandssysteem maar wordt door niemand meer ondersteund en zal uiteindelijk een stille dood sterven, vandaar de keuze voor ext3.
- We gebruiken ook altijd de 64-bit versie van SLES zodat we de beschikking hebben over al ons ramgeheugen.

7.1.4. Prestatiefactoren voor het RAID-systeem

Ook op hardware-gebied is optimalisatie mogelijk. En dan vooral in de RAID-controller kunnen we veel opties instellen die een grote invloed hebben op de prestaties. Let er wel op dat ons doel hier niet was om de KMO na te bootsen, maar we wilden een zo snel mogelijk systeem om de CPUs te kunnen testen.

- RAID-level: zoals algemeen bekend is RAID-0 het snelst.
- Stripe Size: standaard op 64KB.
- Disk Cache zetten we op disabled, omdat hier anders de resultaten kunstmatig zullen worden beïnvloedt. We hebben dit echter uitgemeten voor Sysbench en de impact van deze is redelijk beperkt.
- Disable BGI (Background Initialization): zetten we op Yes, dit betekend dat de initialisatie niet zal plaatsvinden terwijl de tests draaien.
- Write Policy: we hadden geen batterij om dit te kunnen aanzetten, maar via de optie BadBBU kunnen we deze toch forceren. Dit is zeker iets dat voor de KMO niet aan te

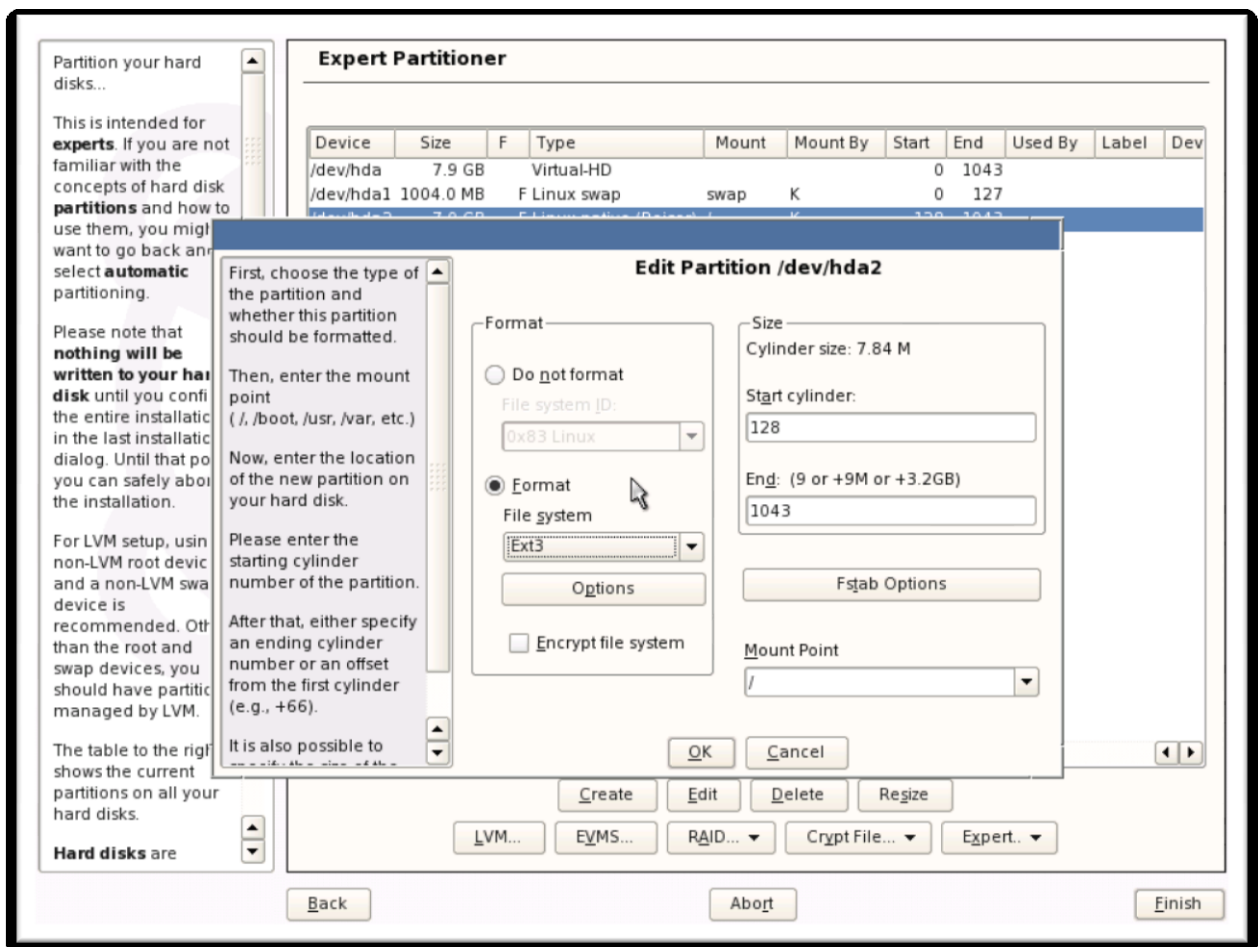
raden is, want dit betekent dat als de server uitvalt alles wat in het geheugen van de RAID-controller zat verloren gaat.

- I/O Policy: hiermee stellen we in wanneer er gecached wordt, met de optie Direct cachen we wanneer een request 2 keer voorvalt.

7.2. Concreet: opzetten van een test met Sysbench

7.2.1. SLES10 installatie

We beginnen uiteraard met de SLES installatie, we gaan hier voor het gemak uit van een virtuele installatie onder Hyper-V, maar dit is gelijklopend voor zowel native als andere virtuele tests. Tijdens deze installatie veranderen we het reiserfs bestandssysteem naar ext3.



Figuur 11: SLES10 installatie: we veranderen het bestandssysteem van reiserfs naar ext3

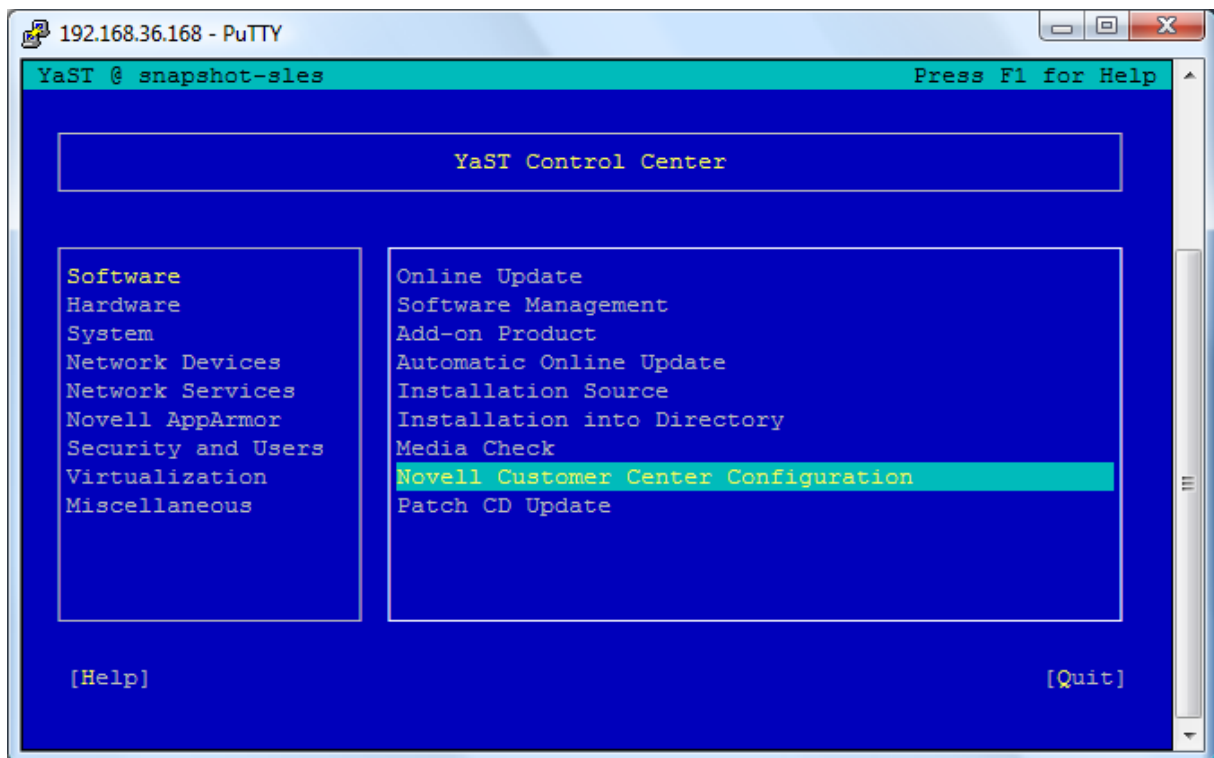
Voor de rest laten we SLES volledig installeren en de volgende actie gebeurd als we de eerste keer ingelogd zijn. We openen hier een terminal en typen de volgende regel in:

```
# vim /etc/inittab
```

Hier zoeken we de volgende regel: `id:5:inittdefaul:` en vervangen het cijfer 5 door het cijfer 3.

Daarna dienen we nog de kernel te updaten, daarvoor moeten we het systeem registeren met een activatiecode die we gratis van internet kunnen halen. We starten YaST op (`# yast`) en kiezen voor “Novell Customer Center Configuration”, hier kunnen we onze versie van SLES Registeren en daarna kunnen we (ook in YaST) kiezen voor “Online Update”. Als we dit

enkele keren doen zal de kernel volledig geupdate zijn naar versie 2.6.16.54-0.2.5. Dit kunnen we testen door in de console `uname -r` te typen.



Figuur 78: SLES10: registratie van het systeem waarna we kunnen Online Updaten

7.2.2. MySQL installatie

Daar het een 64bit installatie betreft halen we de RPM-pakketten af van MySQL 5.1.23 voor SLES10 x86_64. Voor Sysbench hebben we niet alleen client & server nodig, maar ook de shared libraries en voor de compile van sysbench hebben we ook het developers pakket nodig van MySQL. In totaal installeren we MySQL dus met deze 4 commandos:

```
# rpm -i MySQL-client-community-5.1.23-0.sles10.x86_64.rpm
# rpm -i MySQL-server-community-5.1.23-0.sles10.x86_64.rpm
# rpm -i MySQL-devel-community-5.1.23-0.sles10.x86_64.rpm
# rpm -i MySQL-shared-community-5.1.23-0.sles10.x86_64.rpm
```

Daarna dienen we het root wachtwoord aan te passen van MySQL:

```
# mysqladmin -u root password "password"
```

En het large configuratiebestand te kopiëren:

```
# cp /usr/share/mysql/my-large.cnf /var/lib/mysql/my.cnf
```

Zoals eerder besproken moeten we nu nog het startup-script van MySQL aanpassen zodat de MySQL daemon gebruik maakt van 2 CPUs en niet van allen. Dit script is te vinden in `/etc/init.d/mysql`. We openen dit script met `# vi /etc/init.d/mysql` en zoeken regel 307:

LETOP: voor AMD CPUs is dit commando anders dan voor Intel CPUs.

We voegen voor regel 307 het volgende commando toe:

Voor Intel: *taskset --cpu-list 0-3 -- \$binddir/...*

Voor AMD: *numactl --physcpubind=0-3 -- \$binddir/...*

Nu moeten we gewoon nog MySQL herstarten zodat deze op de 2 CPUs draait:

```
# /etc/init.d/mysql restart
```

7.2.3. Sysbench installatie

Deze moet gecompileerd worden, maar daarvoor is het gcc pakket nog nodig. We installeren deze door de SLES10 DVD in te steken en in de console *# yast -i gcc* te typen.

Daarna kunnen we de Sysbench 0.4.8 broncode afhalen en compileren als volgt:

```
# tar xzf sysbench-0.4.8.tar
# chmod -R 777 sysbench-0.4.8/
# chown -R root:root sysbench-0.4.8/
# cd sysbench-0.4.8/
# ./configure
# make && make install
```

7.2.4. RAID-preparatie

Tot slot moeten we nog de externe RAID-schijf verbinden en prepareren om de MySQL database-files erop te zetten, we veronderstellen even dat deze schijf */dev/sdb* heet.

```
# fdisk /dev/sdb
"n" (nieuwe partitie), "p" (primary), "1" (partitie #1), 2xEnter (defaults), "x" (expert mode),
"b" (startblokken specificeren), "1" (partitie #1), "64" (aligneren), "r" (normale mode),
"t" (partitie type), "1" (partitie #1), "83" (= ext3 volume), "w" (opslaan en fdisk afsluiten)
Daarna formateren we deze partitie als ext3: # mkfs.ext3 /dev/sdb1.
```

Om de datafiles te kopiëren naar deze partitie moeten we ze tijdelijk mounten, daarna alles kopiëren en dan terug mounten op de plaats waar de originele stonden. Op dat ogenblik is alles klaar voor de test.

```
# /etc/init.d/mysql stop
# mount /dev/sdb1 /mnt/
# cp -rp /var/lib/mysql/* /mnt/
# rm /mnt/*log*
# umount /dev/sdb1
# mount /dev/sdb1 /var/lib/mysql/
# chown -R mysql:mysql /var/lib/mysql/
# /etc/init.d/mysql start
```

7.2.5. Een test starten

Hiervoor bestaat een script maar hier wordt uitgelegd wat er allemaal gebeurt bij het uitvoeren van dit script.

We beginnen met het aanmaken van de sbtest database:

```
# mysql -uroot -p$PASS -e "create database sbtest"
```

Zoals eerder gezegd moeten we eerst de test voorbereiden (prepare), dit kan met dit commando:

```
# sysbench --test=oltp --oltp-table-size=1000000 --mysql-table-engine=innodb --mysql-socket=/var/lib/mysql/mysql.sock --mysql-user=root --mysql-password=PASS prepare
```

Daarna kan de test zelf uitgevoerd worden (eventueel meerdere keren maar met verschillende aantal van threads). Let erop dat we dit commando ook vastpinnen aan een CPU omdat dit virtueel ook niet anders zou zijn. (numactl wordt op een Intel CPU vervangen door taskset).

```
# numactl --physcpubind=0 -- sysbench --num-threads=$x --max-requests=25000 --test=oltp --oltp-read-only=off --mysql-table-engine=innodb --oltp-table-size=1000000 --mysql-socket=/var/lib/mysql/mysql.sock --mysql-user=root --mysql-password=PASS run > test.txt
```

De output wordt hier dus wel opgeslagen in een bestand zodat we de resultaten niet kwijt zijn. Na afloop van de test kunnen we een cleanup doen met het volgende commando:

```
# sysbench --test=oltp --mysql-socket=/var/lib/mysql/mysql.sock --mysql-user=root --mysql-password=PASS cleanup.
```

Tot slot nog de sbtest database terug verwijderen:

```
# mysql -uroot -pPASS -e "drop database sbtest"
```

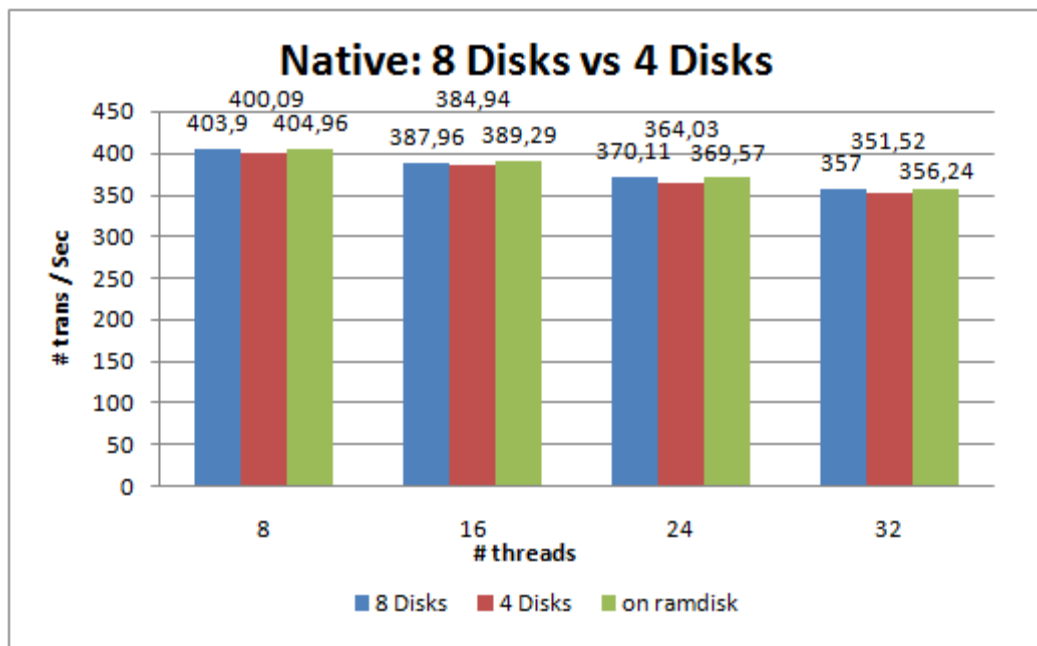
7.3. Testresultaten

Deze dienen als hoofdzaak om onze eerdere best-practices te staven, en om onze tuning parameters te bewijzen. Nieuwe informatie wordt hier niet gegeven.

7.3.1. Bewijzen, algemene tests

8 of 4 schijven

Om te bewijzen dat de limiet niet bij het RAID-systeem zit en we wel degelijk op de CPU prestaties leunen hebben we eens een RAID-systeem verbonden van 4 schijven in plaats van 8 en hebben we eens op de interne RAM getest.

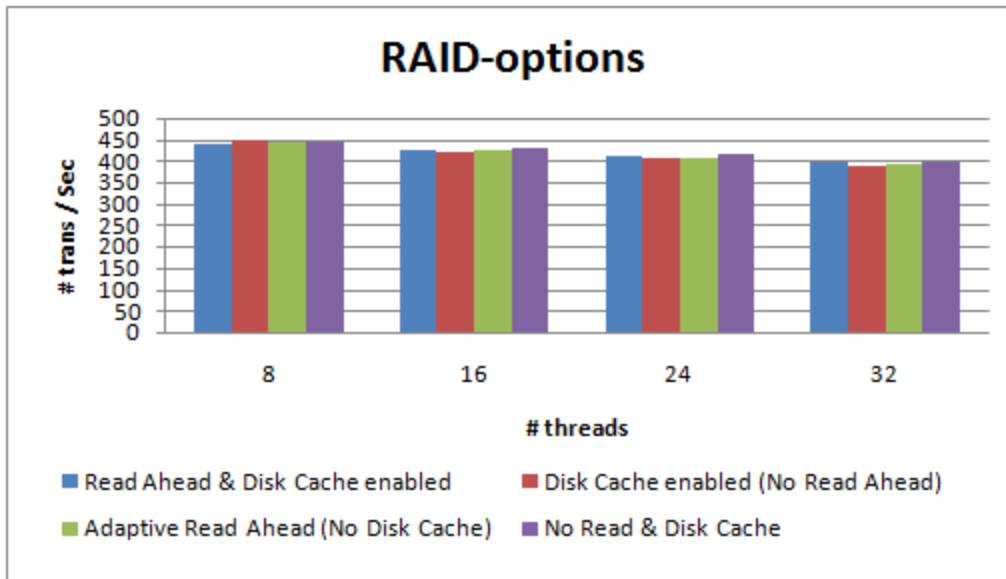


Figuur 79: Native test: 8 schijven, 4 schijven of intern geheugen

Het verschil is onderling nooit meer dan 1,5% en dus verwaarloosbaar. Enkel als we het aantal CPUs verhogen zien we een zwaar verschil. Bijv 8 CPUs geven een verschil van 117%.

RAID-settings

De eerder genoemde instellingen op de RAID-controller die volgens ons weinig impact hebben op deze OLTP-benchmark zijn Disk Cache & Read Cache. Om dit te staven hebben we de test herhaald met de opties uit- of aangeschakeld.



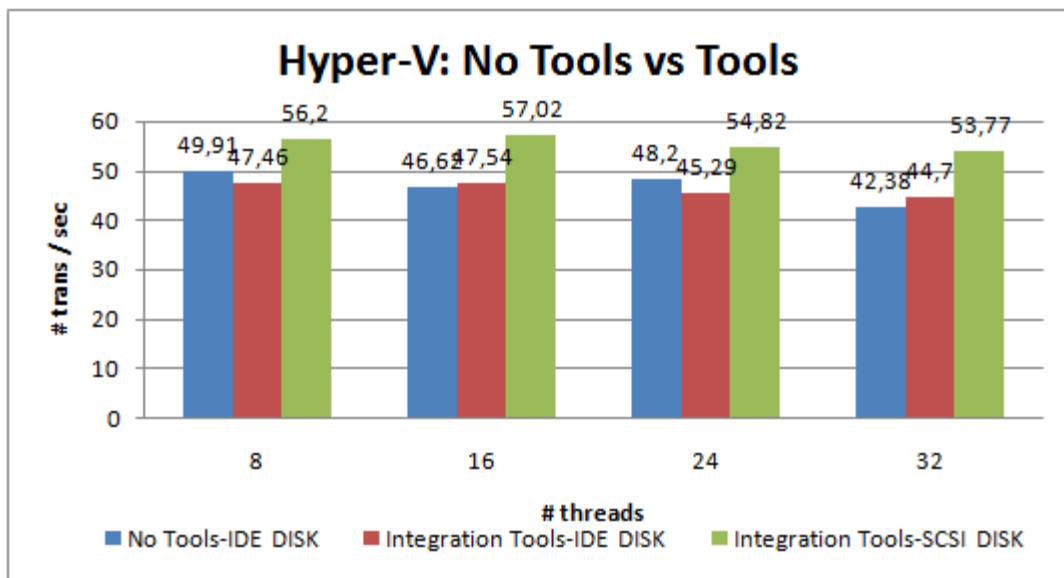
Figuur 80: RAID-test: Disk & Read Cache

7.3.2. Barcelona tests

Hyper-V Integration Tools en VMware ESX Tools

We bekijken eerst Hyper-V. Alle tests zijn uitgevoerd zonder installatie van de Integration Services, maar we bekijken toch eerst eens de impact die zo'n paravirtualisatie in zich heeft.

Let wel! Door een kleine bug in het patchen van de Xen kernel kan deze slechts 1 vCPU aanspreken, dus we kunnen deze niet vergelijken met de verdere resultaten die allemaal op 2 CPUs werden gedaan.



Figuur 81: Hyper-V test met Integration Tools

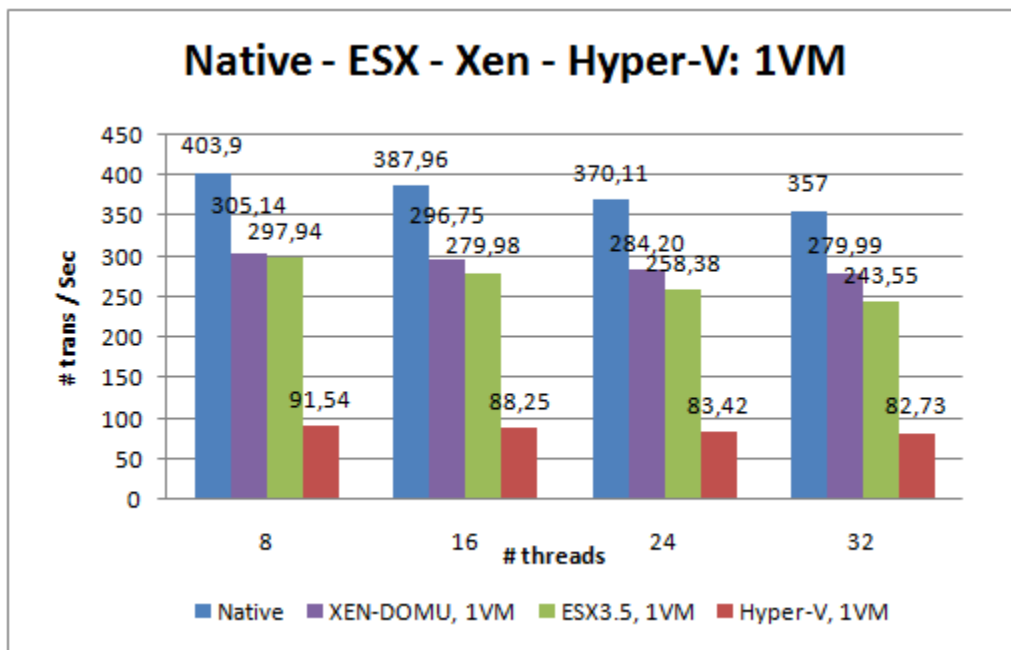
Uit bovenstaande test blijkt dat de tools op zich zeker geen impact hebben als we niet de controller overschakelen naar een SCSI Controller. Dit is logisch want de IDE controller wordt niet geparavirtualiseerd, de SCSI Controller wel.

Het verschil tussen de eerste en tweede reeks bedraagt gemiddeld 1,1% terwijl dat tussen de eerste en derde reeks 18,5% is.

Voor VMware geldt een ander verhaal, hier zijn de tools louter driverupdates voor de grafische of netwerk-kaart. Of het toevoegen van enkele automated scripts. Dit blijkt ook uit de resultaten: het verschil valt binnen de foutmarge en bedraagt: 0,1%.

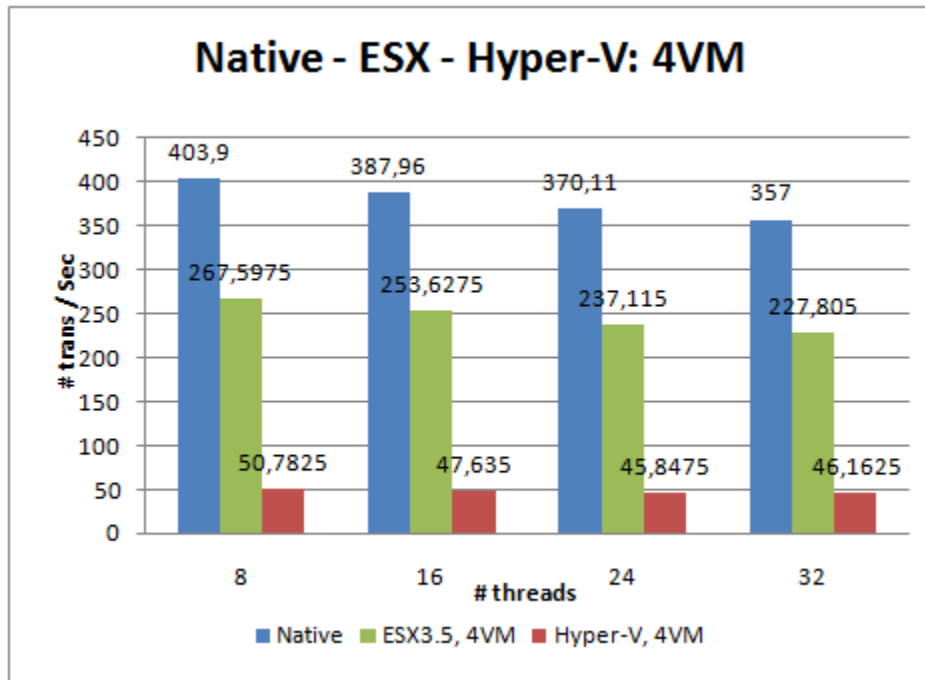
Barcelona: platform-verschillen

We zetten even (ter info) de resultaten tegenover elkaar van een Native situatie, ESX situatie, Xen domU situatie & een Hyper-V situatie. Zowel met 1 Virtuele Machine als 4 Virtuele Machines. Wegens tijdsnood hebben we enkel voor de Barcelona-machine Xen resultaten.



Figuur 82: Barcelona-tests: 1 virtuele machine

De ESX overhead ten opzichte van native is hier 28,9% verlies, voor Xen is er maar 23,2% verlies en voor Hyper-V bedraagt dat maar liefst 77,2% verlies.



Figuur 83: Barcelona-tests: 4 virtuele machines

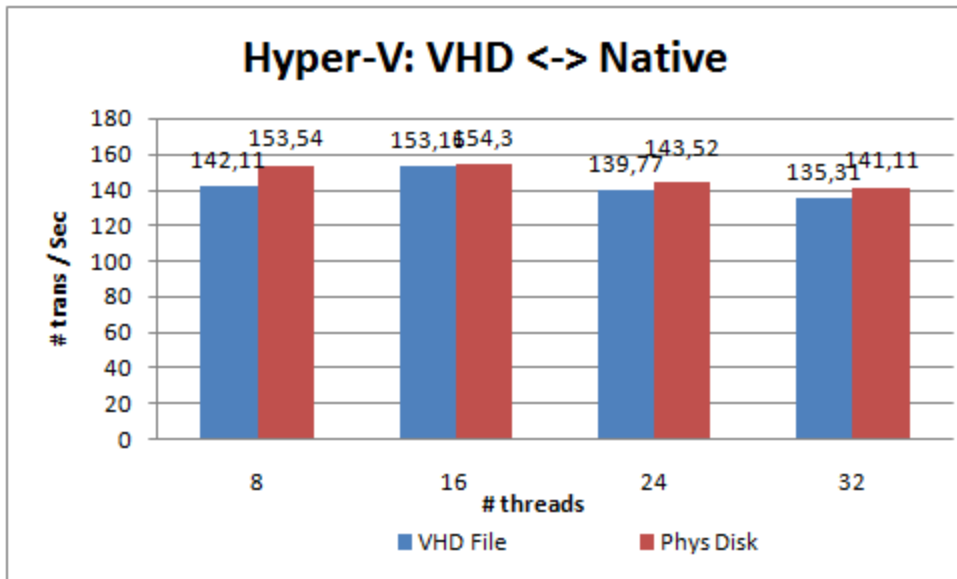
Hier is de ESX overhead ten opzichte van native 35% verlies en bij Hyper-V is dat verlies gestegen naar 87%.

Hyper-V ligt duidelijk nog niet in een definitieve plooi.

7.3.3. Clovertown tests

Hyper-V: Fysieke schijf tov virtuele schijf

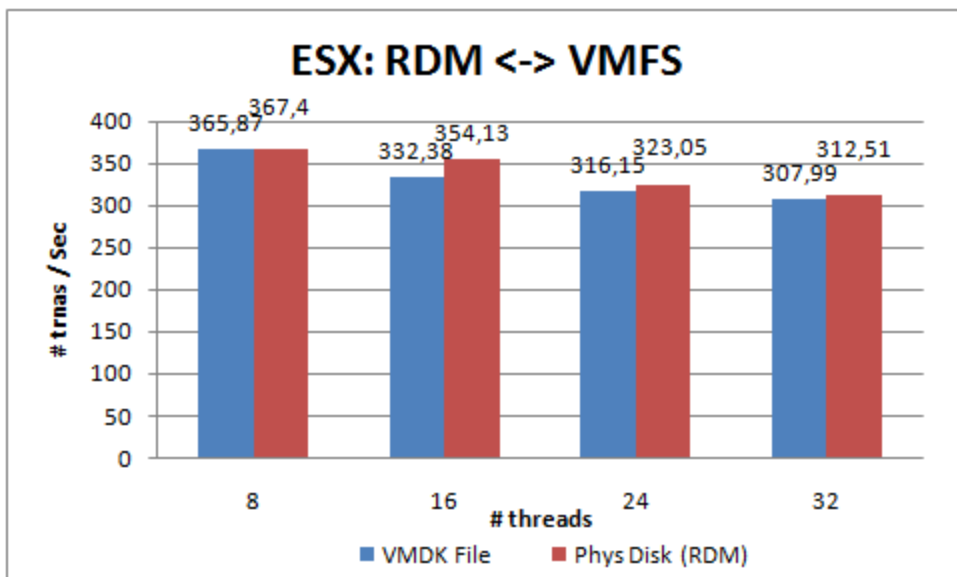
Er zou een prestatie winst moeten zijn als we in Hyper-V de schijf volledig aan een VM toekennen in tegenstelling tot het gebruik van vhd-bestanden als schijven. Er is minder controle (van bijv. snapshots) en dit zou de toegang iets versnellen. Als we dit uitmeten zien we echter een beperkte, maar nog steeds meetbare prestatiewinst van 3,8%.



Figuur 84: Hyper-V resultaten van fysieke en virtuele schijven

ESX: Fysieke schijf tov virtuele schijf

Bij ESX bestaat Raw Disk Mapping (RDM) en in theorie zou dit beter moeten presteren dan de virtuele schijf omdat de virtuele machine via RDM rechtstreekse toegang heeft tot de schijf. Dit in tegenstelling tot een virtuele schijf waar alle commandos via de VMM moeten passeren. We testen dit even in een situatie met maar 1 virtuele machine. Zodat enkel die overhead gemeten wordt.



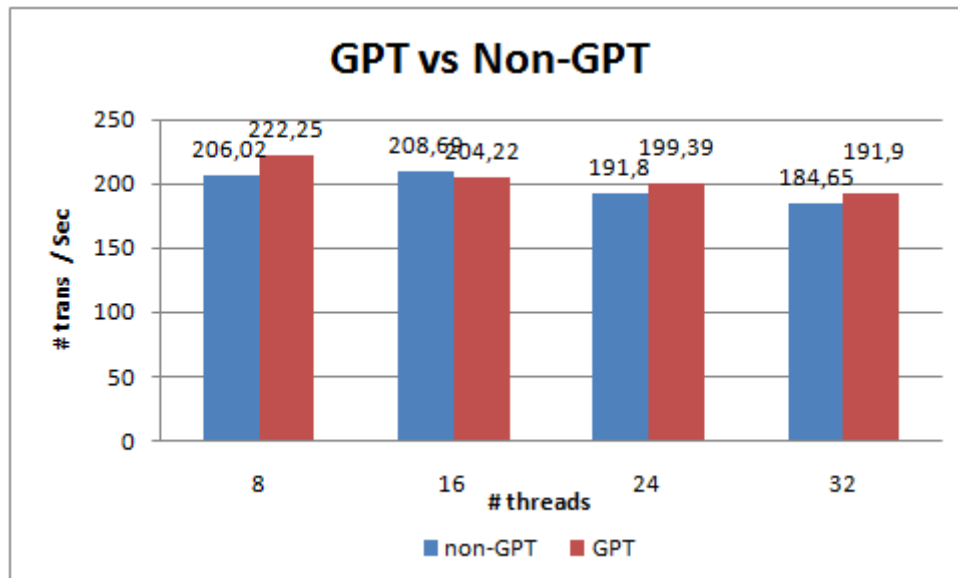
Figuur 85: ESX resultaten van fysieke en virtuele schijven

Ook hier is er zeer kleine prestatiewinst als we een schijf via RDM benaderen. Het verschil is 2,6%

7.3.4. Harpertown tests

GPT tov MBR

Onder Windows kunnen we bij het initialiseren en partitioneren van een schijf kiezen of we dit volgens de GPT of MBR manier willen doen. Op virtueel niveau onderzochten we even de impact door een test te doen als het vhd bestand op een GPT partitie staat, of anderszijds op een gewone MBR partitie.

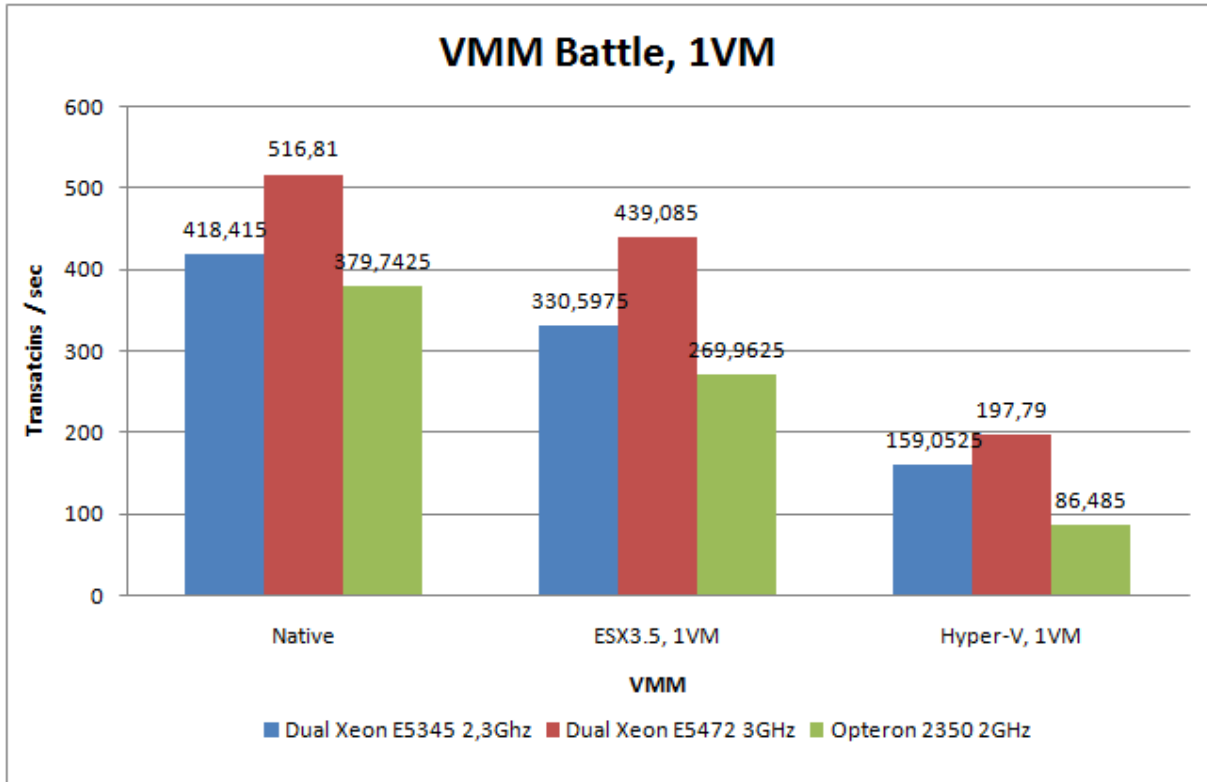


Figuur 86: Harpertown-test: GPT of MBR

Er is een gemiddelde prestatiewinst van 3,3% als we met GPT werken, in de praktijk valt dit dus te verwaarlozen.

7.3.5. Platform vergelijking

Wat komt er nu te voorschijn als we verschillende resultaten van de machines tegenover elkaar zetten. We nemen eerst de resultaten met 1 virtuele machine. We werken telkens met de gemiddelde waardes. Om de vergelijking beter te kunnen maken gebruiken we niet de codenamen, maar hun volledige namen, want de klok-frequentie geeft namelijk ook een grote impact op deze Sysbench tests.



Figuur 87: Server Battle: 3 machines en 3 platformen naast elkaar

Als we de platformen vergelijken zien we dat Hyper-V veruit de slechtste is, wat kan te wijten zijn aan het feit dat het nog maar een Release Candidate 0 en dus nog geen finale versie is. ESX doet het beter maar er zit gemiddeld toch een 21% prestatieverlies op.

Als we tenslotte ook eens de hardware-platformen naast elkaar leggen wordt het moeilijker doordat ze alledrie op een andere frequentie geklokt worden.

Als we even die frequentie zouden uitschakelen (dit ter info), kunnen we ze misschien beter vergelijken. De resultaten worden omgerekend naar 3GHz en dan merken we dat de 3 platformen zeker aan elkaar gewaagd zijn en nooit meer dan 5% van elkaar verschillen qua prestaties.

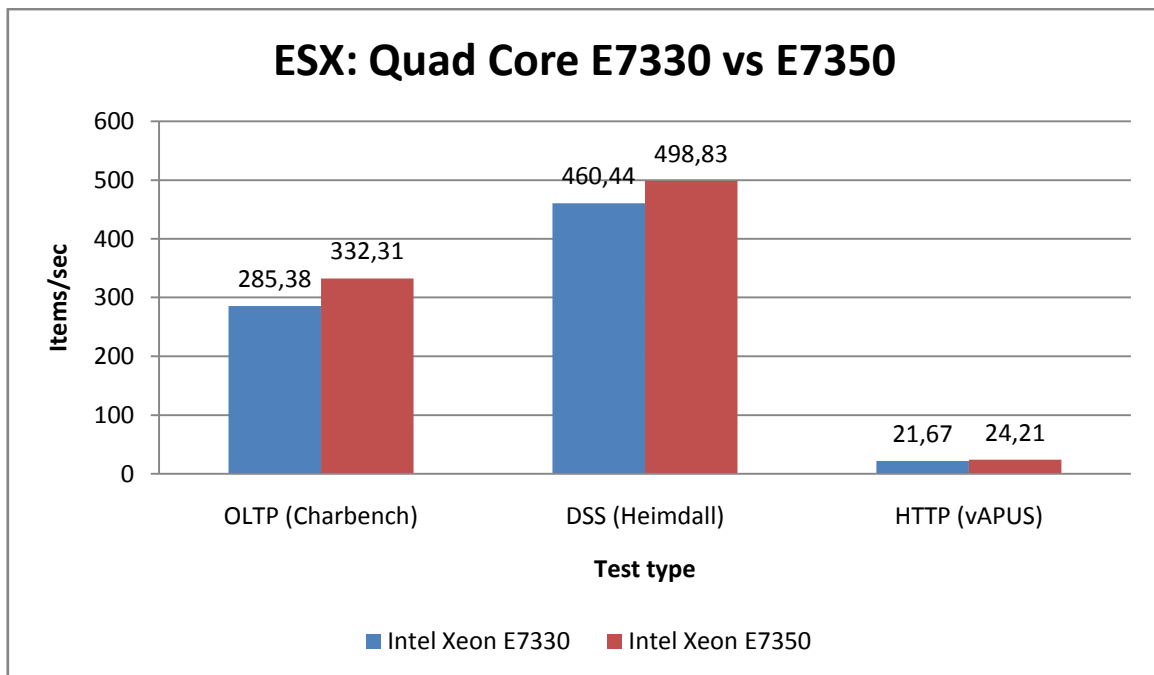
7.3.6. Real World testscenario's

Er werd een volledig nieuw hardware platform voor deze tests gebruikt, dit omwille van het feit dat we op die manier de beschikking hadden over machines met 16 cores. Dit zijn 4 sockets met in elke socket een Quad-Core CPU geprikt. Het enig verschil tussen de machines onderling is de CPU. Er waren 3 CPUs voorhanden:

Tabel 3: Overzicht Quad Core Processors

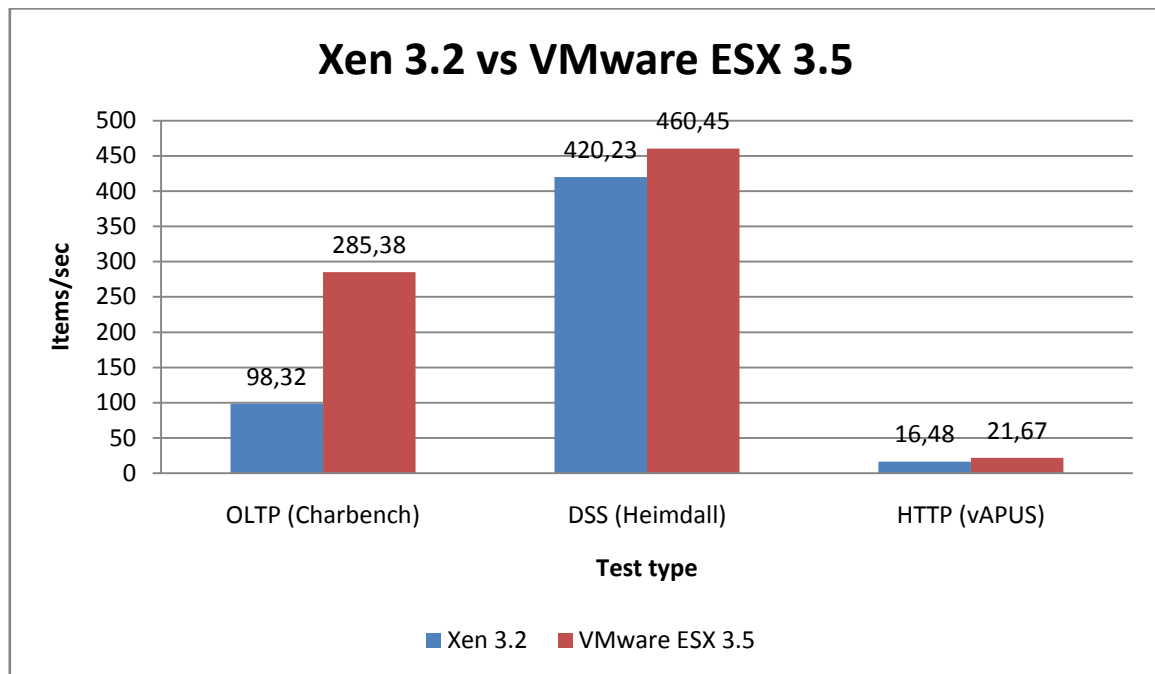
Specificatie/CPU	Intel Xeon E7330	Intel Xeon E7350	AMD Opteron 8356
Kernfrequentie	2400MHz	2930MHz	2300MHz
Productieproces	65nm	65nm	65nm
FSB / HyperLink speed	1066MHz	1066MHz	1008MHz
L2 / L3 Cache	2 x 3MB L2	2 x 4MB L2	2MB L3

Het verschil tussen de 2 Intel processors ligt voornamelijk in de klokfrequentie maar ook in de grootte van de caches. Na het uitmeten kunnen we vaststellen dat de gemiddelde impact bij ESX zo'n 21% is, uiteraard ten voordele van de E7350. Vooral de webtest leunt sterk op de CPU snelheid, met als resultaat een prestatiewinst van 42%.



Figuur 88: Real World scenario: E7330 vs E7350

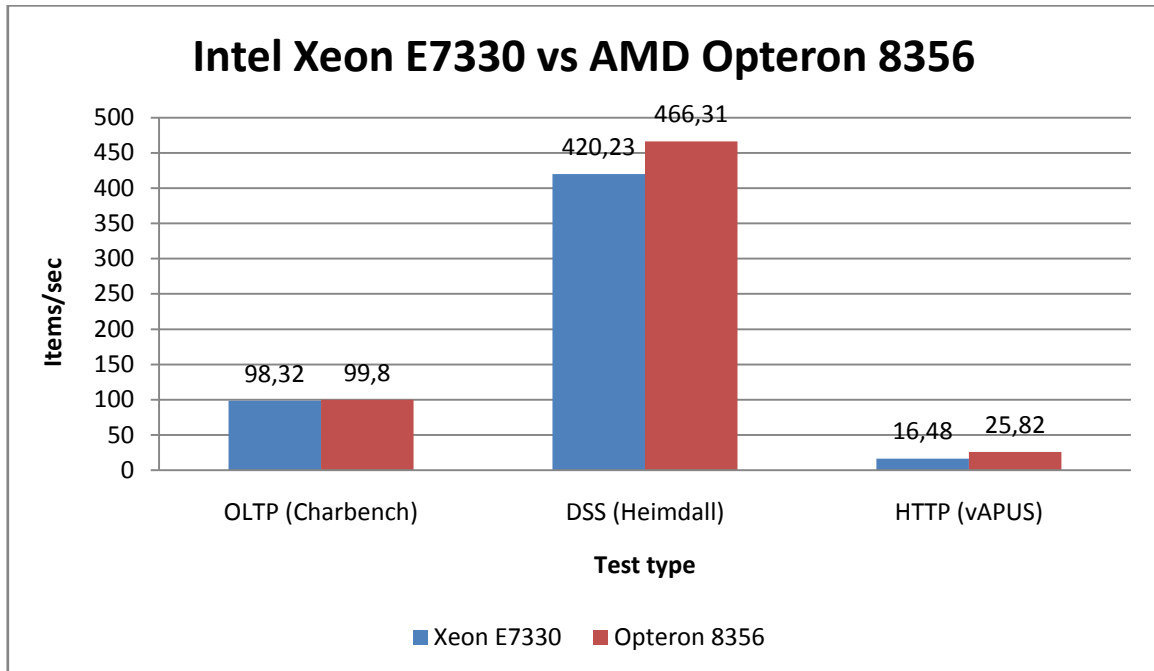
Verder is ook het onderscheid tussen Xen 3.2 en VMware ESX 3.5 belangrijk, ook op deze nieuwe platformen hebben we dit getest, en wel met het Real World testscenario. De gebruikte CPUs waren telkens Intel Xeon E7330. ESX doet het altijd beter en vooral in de OLTP tests weet ESX zelfs een voorsprong van 65% te behalen. Er zijn echter wel situaties waar Xen minder slecht presteert, zoals de webtest (verlies: 8%).



Figuur 89: Real World Scenario: Xen vs VMware

Uiteraard kan Xen zijn paravirtualisatie kaart niet uitspelen, aangezien het hier virtuele Windows systemen betreft, rekent Xen op zijn HVM. De reden dat hier voor Xen 3.2 werd gekozen is dat deze officieel ondersteuning biedt voor de Full Virtualization van Windows systemen. Er werd verder uitgemeten dat Windows op Xen 3.2 inderdaad reeds naar behoren werkt, maar de Binary Translation van ESX blijft een behoorlijk voordeel hebben tegenover Xens HVM. De OLTP test schrijft zijn data relatief snel weg naar de schijven en daardoor kan ESX sterk de herhaalde identieke instructies van de VM hergebruiken uit zijn Translator Cache. Xen rekent op de VMCS in de Intel VT CPU om terug te schakelen naar de hypervisor en dit is zeer kostbaar qua klokpulsen. DSS en vooral de webtest hebben daar veel minder last van.

Het voordeel van Nested Page Tables kunnen we nu uitmeten doordat we de beschikking hebben over de Xeon E7330 zonder Extend Page Tables en de nieuwe Opteron 8356 die wel Nested Page Tables ondersteund. Beide CPUs zijn ongeveer gelijkaardig geklokt en dus behoorlijk vergelijkbaar met elkaar. Als hypervisor werd gekozen voor Xen 3.2 omdat deze officieel ondersteuning biedt voor NPT.



Figuur 90: Real World Scenario: E7330 vs Opteron 8356

Zoals eerder besproken hangt de prestatiewinst af van het type workload. De webtest blijkt het grootste voordeel te halen uit NPT met een winst van 49%, de vele geheugen interacties (I/O operaties) hebben daar ongetwijfeld mee te maken. De OLTP test is eerder transactioneel en rekt vooral op de kracht van de schijven en het aanspreken van ervan, er gebeuren weinig geheugenoperaties en dus kan NPT weinig invloed uitoefenen bij deze test. De Heimdall test bevindt zich ergens in het midden, omdat deze zowel schijf als geheugenbeheer nodig heeft.

8. Conclusie & besluiten

Door te kiezen voor deze masterproef ging er een wereld open die rechtstreeks te maken had met zowel nationale als internationale bedrijven. Het Sizing Server project is een koepel waarbinnen onder andere deze masterproef tot een goed einde werd gebracht. Het is echter zoveel meer dan dat.

Eén van de pijlers van het TETRA-project is namelijk een gebruikerscommissie, dit zijn meerdere bedrijven die technologische input hebben op het onderzoek en die gedeeltelijk financiële steun verzorgen. In ruil voor een kleine, financiële vergoeding doet het project dan onderzoek op aanvraag van deze bedrijven. De resultaten van dit onderzoek zijn dan gegarandeerd meteen inzetbaar binnen de aangesloten bedrijven en leveren de projectmedewerkers meteen kennis & ervaring op voor volgende projecten en keuzes.

Dit alles in overweging genomen wordt stilaan de impact van een thesis (of masterproef) als deze duidelijk.

Voor een groot, internationaal bedrijf is SAN een vanzelfsprekend iets geworden. Enkele Fibre Channel switchen, honderden kabels, storage racks en servers uitgerust met Fibre Channel kaarten zijn een veel voorkomende zaak in de server rooms van deze grote & rijke bedrijven. Hun inkomsten komen dan ook vanuit verschillende uithoeken van de wereld. Een website die 24 uren per dag moet bereikbaar zijn, een applicatie die bij een uitval klanten teleurstelt. Dit zijn zeer kostbare dingen voor een bedrijf zo groot als Google of British Telecom, SAN wordt dan al gauw onmisbaar. Ook voor een KMO is Shared Storage zeker de moeite waard, maar wordt hier dan van weerhouden door de torenhoge prijzen van een Fibre Channel installatie.

Sinds de komst van gigabit ethernet en iSCSI is hier gelukkig verandering in gekomen. Met deze masterproef werd bewezen dat mits een paar kleine optimalisaties en aanpassingen iSCSI in 90% van het standaard gebruik minstens even flexibel & snel is als Fibre Channel. En het wordt nog beter!

Deze zomer staat 10-gigabit ethernet over koper op het programma om op de markt te komen en dit zal de grens tussen Fibre Channel & iSCSI volledig wegnemen. De eerste adapters zijn besteld en reeds onderweg naar het Sizing Server project, waar vele kritische ogen hen opwachten.

Wat eerst begon als een desktop hype groeide uit tot een technologie die op de server markt voor grof geld over de toonbank gaat. Volledige serverparken zijn gevirtualiseerd en als er zich een ramp voordoet in de ene stad, zal een ander serverpark het werk gewoon overnemen. Een gebruiker merkt geen verschil en laat dát nu net de definitie van virtualisatie zijn. Voor een KMO zijn meerdere serverparken en de vele VMware licenties die daarmee gepaard gaan nog wat te hoog gegrepen. Maar dit wil niet zeggen dat deze KMOs geen virtualisatie kunnen toepassen. Een secundair doel van deze masterproef was het voorstellen van oplossingen voor deze KMO's die ook een vorm van hoge beschikbaarheid willen nastreven. Die ook een internationale website willen aanbieden of een applicatie die ten alle tijde bereikbaar moet zijn.

Veel mensen denken echter dat virtualisatie voor alles een oplossing biedt, met een prestatieverlies van slechts 5% zou er geen reden zijn om niet aan virtualisatie te doen. Niets is echter minder waar. Niet alleen wordt die 5% enkel in bepaalde (zeer beperkte) workloads gehaald, het aantal virtuele systemen die op een fysieke machine draaien speelt ook een grote rol in de onderlinge prestaties. Verder zijn er simpelweg applicaties die niet virtueel mogen/kunnen draaien, zoals rendermachines of zware ontwikkelingstools.

Xen mag dan wel vrije software zijn, voor bedrijven die op Windows draaien lijkt dit nog niet meteen een perfecte mogelijkheid. Echter, de oplossing komt misschien wel uit een hele andere hoek: namelijk van Microsoft zelf. Met hun nieuw server-platform en het feit dat ze een alliantie hebben met Citrix, kunnen ze vriend en vijand nog verrassen. Hyper-V is nog verre van perfect, maar voor een Release Candidate zijn de mogelijkheden toch reeds uitgebreid. Functies als Microsoft clustering, High Availability & Load Balancing trekken zeker de aandacht en smeken om verder onderzoek. Iets wat zeker zal gebeuren, een serverpark dat beheerd wordt door Hyper-V lijkt nog veraf maar er zijn zeker stappen in die richting.

De impact van het gebruik van Shadow Pages wordt groter naarmate er meer geheugenintensieve operaties nodig zijn. Zo speelt de cache van de CPU ook een grote rol, hoe minder cache, hoe meer er naar het geheugen moet worden gegaan en dit vereist een frequentere vernieuwing van de Shadow Pages. Applicaties die veel geheugentoeegang vereisen (bijvoorbeeld door slechte verdeling van de threads) zullen dus zeer sterk lijden onder de Shadow Pages en hebben veel baat bij grotere CPU-caches.

Belangrijk is om op te merken dat hardware virtualisatie op zich hiervoor geen hulp is, door de toevoeging van een VMCS in de CPUs wordt het geheugenbeheer niet uit handen genomen van de hypervisor.

Nested Page Tables kunnen hier echter wel een grote hulp zijn, door de TLB gevoelig te vergroten en met de toevoeging van de extra hardware tag nemen ze veel software-werk af. TLB-flushes bij context switches (schakelen tussen VMs) zijn niet meer noodzakelijk. De resultaten liegen er niet om; bij geheugenintensieve applicaties worden prestatiewinsten tot 49% gemeten.

9. Case Studies

9.1. Case Study 1: MCS VMware uitbreiding

CASESTUDY MCS VMWARE UITBREIDING

www.mcs.be – info@mcs.be – Sneeuwbeslaan 20, B-2610 Antwerpen

Tel: +32/(0)3 829 04 95



VOORSTELLING BEDRIJF

MCS of Management Consultancy Services werd opgericht in 1989 door Marcel Eeckhout en is een Naamloze Vennootschap. Zijn klantenbestand bevat voornamelijk klanten uit België en Nederland. Echter een aantal jaren terug is MCS partnerships aangegaan met internationale partners als ISS uit Denemarken en Siemens uit Duitsland. Op dit moment is MCS zelfs actief in 29 landen.

ACTIVITEITEN BEDRIJF

MCS biedt zijn klanten software en consultancy oplossingen aan betreffende hun secundaire processen om zo inzicht, alertheid, flexibiliteit en efficiëntie te verhogen. We doen dit met een geïntegreerde aanpak die de hoogste Return On Investment garandeert. Reeds vanaf de oprichting spitst MCS zich naast Facility Management tevens toe op Real Estate. Onze missie, innoveren en continu streven naar meer kwaliteit, weten ook onze klanten te waarderen.

Bedrijfsmissie: “Facility Management begint bij Real Estate. Facility Management heeft maar zin als het management wordt. Maar mist zijn doel als het blijft hangen bij facilities”

VOORSTELLING CASE

PROBLEEMVOORSTELLING

MCS wil hun infrastructuur opwaarderen en voorbereiden voor uitbreiding door te gaan werken met virtualisatie. Op dit moment bestaat reeds een ESX server op hun netwerk en bijgevolg lijkt het logisch dat de keuze voor het platform valt op VMware ESX.

Waarom virtualisatie

Pro:

- Server consolidatie: dit is meer dan de server op zich, maar ook de harde schijf, het geheugen, netwerkkaarten, management ...
- Backup / Restore & uptime: servers worden nu bestanden op een server en deze zijn veel makkelijker handelbaar
- Flexibiliteit: een server kan vrij eenvoudig uitbreiden qua CPU en geheugen. Een virtuele server kan ook hardwarematig veranderen volgens een tijdschema

- Zeer eenvoudig om (al dan niet tijdelijk) een nieuwe (test-)server bij te creëren

Cons:

- Consolidatie is niet eindeloos: geheugen wordt duur of na verloop van tijd uitverkocht
- Als een fysieke server uitvalt liggen meteen vele virtuele servers uit
- VMware draait op unix en kan specifieke kennis vereisen bij het oplossen van diepgaande problemen
- Prijs: licentiekosten voor ESX zijn zeer duur, de duurste vorm (Enterprise editie) is daardoor niet haalbaar

Huidige situatie

Momenteel is er één server “srmcs10” aanwezig, deze werd als kennismaking opgezet maar wordt meer en meer voor productiedoelinden gebruikt.

SRVMCS10 (HP Server)

CPU: 2 Dual Cores

Geheugen: 8 x 1GB

Harde schijf: RAID 5 3 x 146GB 10k RPM met write cache

ILO2 Remote Management

Netwerk: 2 x ethernet

Software: ESX 3 Starter

Voorstel voor toekomst

Een iSCSI SAN netwerk (= aparte server) plus een nieuwe ESX server, dit brengt het totaal op 2 ESX Servers en een storage server.

SRVMCS12 (HP 380 G5)

CPU: 1 Quad Core

Geheugen: 2 x 4GB + 1x2GB + 4 vrije sloten

Harde schijf: RAID 5 3x146GB 10k RPM met write cache

ILO2 Remote Management

Netwerk: 4x ethernet

Software: ESX 3 Standard

De eerste server “srmcs10” zou dan een uitbreiding krijgen naar ESX 3 Standard, een geheugenuitbreiding naar 12Gb (4x1GB + 2x2GB + 2 vrije sloten) en een extra netwerkkaart zodat het totaal op 4 komt.

Het grootste voordeel van een 2^{de} ESX server is het feit dat de virtuele servers kunnen migreren van de ene naar de andere machine, als een fysieke machine dan uitvalt kunnen de virtuele servers opstarten op de andere machine. Hiervoor is echter gedeelde opslag

nodig (SAN) om de schijfbestanden op te slaan. De enige oplossing voor SAN is Fibre Channel (te duur voor MCS) of iSCSI. Vandaar de nood aan een iSCSI Server.

OPLOSSINGSVOORSTELLING

VMWARE ESX 3.5

Een overzicht van de verschillende ESX versies en hun ondersteuning:

Tabel 4: ESX versies en hun ondersteuning.

	ESX Foundation (voorheen: Starter)	ESX Standard	ESX Enterprise
ESX Server	✓ • NAS or Local Storage • Only 4 CPU-sockets & 8GB Memory	✓	✓
VMFS & Virtual SMP	✓ • Local storage only • No cluster file system	✓	✓
VirtualCenter Agent	✓	✓	✓
Consolidated Backup	✓	✓	✓
Update Manager ¹	✓	✓	✓
VMware HA ¹		✓	✓
VMotion ¹ Storage VMotion ¹ VMware DRS ¹			✓
Prijs	\$995	\$2995	\$5750

¹: Voor deze optie is een VirtualCenter Server met aparte licentie vereist (kostprijs vanaf \$6044)

Vanaf ESX 3.5 wordt de naam ESX Starter vervangen door ESX Foundation. Echter deze goedkoopste versie is beperkt tot NAS of lokale opslag en bijgevolg geen optie als we 2 ESX Servers willen inzetten. Of willen kunnen migreren.

ESX Enterprise is de duurste versie met ondersteuning voor High Availability en VMotion maar is omwille van zijn zeer hoge kostprijs \$5750 per licentie ook geen optie.

Dus blijven we bij de ESX Standard, die op beide fysieke hosts van MCS dient te worden geïnstalleerd. Nog een klein overzicht van de opties die bij deze ESX versie zitten:

- VMFS & Virtual SMP: ESX werkt standaard met een eigen ontwikkeld bestandssysteem om zijn zeer grote schijfbestanden te kunnen beheren. Iedere ESX versie werkt met dit VMware File System of VMFS. Virtual SMP is de mogelijkheid om meerdere CPUs te kunnen toekennen aan één en dezelfde VM en het beheer van deze CPUs zagezegd aan het virtuele gastsysteem overlaten. De limiet is op dit moment echter tot 4 vCPUs.
- VirtualCenter Agent, dit de webbrowser die ervoor zorgt dat de Virtual Infrastructure Client kan verbinden met de ESX Host om deze op die manier grafisch te kunnen beheren.
- Consolidate Backup; een set van configureerbare scripts om op gezette tijdstippen backups van bepaalde VMs te kunnen uitvoeren. Dit kan naar USB schijfstations, tapedrives, NAS locaties ...
- Update Manager: Software die controleert of een host up-to-date is en deze desgewenst updaten. Hij zal tevens de (vooraf ingestelde) virtuele machines monitoren en vergelijken met vooraf ingestelde parameters en deze dan automatisch update. Hij houdt ook een snapshot bij van de VM om desgewenst terug te keren naar de status voor de update. Hiervoor is echter de VirtualCenter Server nodig (die dit geheel zal overzien) en omwille van de hoge kostprijs is deze geen optie.
- VMware High Availability: als door een crash of hardware-fout in 1 van de ESX machines een of meerdere virtuele machines onbruikbaar worden zal VMware HA er automatisch voor zorgen dat de betreffende VM opnieuw opgestart wordt op een andere host. Ook deze functie vereist een VirtualCenter Server. Echter als we zelf de VM migreren bij een crash kunnen we de HA manueel doen.

OPSLAG-INFRASTRUCTUUR: iSCSI

Waarom geen Fibre Channel? Een kleine opsomming van de kosten:

Allereerst is er een Fibre Channel storage array, zoals de Promise e310F, deze begint vanaf \$4000 (shopping.yahoo.com). Zonder de schijven. Dit is op zich wel zeer vergelijkbaar met een iSCSI server.

Maar dan komen er nog speciale optische kabels, fibre channel switches en Host Bust Adapters bij.

Een optische LC-LC kabel (3meter) kost € 86 en voor een 10meter versie stijgt dat meteen naar € 123 per stuk (mpl.be) .

Een HP StorageWorks 8ports 4Gb FC Switch kost \$ 7911 (shopping.yahoo.com) terwijl een simpele HBA (LSI SinglePort PCI-X van 4Gb) begint vanaf € 577 (mpl.be).

In het geval van MCS betekend dit een investering van 2 x € 577 + een switch (omgerekend € 5122) + een storage array (omgerekend €2590) + 2 x € 86 = **9038 Euro**.

En dan vergeten we nog de kosten van een geheel nieuwe infrastructuur (kabels leggen) en het feit dat er minder mensen bekend zijn met het Fibre Channel protocol ten opzichte van TCP/IP.

Het wordt echter een heel ander verhaal als we denken aan iSCSI. De bestaande netwerk-infrastructuur kan behouden worden of eventueel geupgrade tot gigabit snelheden. Alle servers hebben (soms meerdere) gigabit-netwerkaarten aan boord.

Voor de iSCSI-structuur is niet elke switch gelijk. Een layer-2 managed switch, zoals de in ons lab gebruikte D-Link DGS-1224, is vereist om met dingen als VLANs of Jumbo-frames te kunnen werken. Beiden zijn best-practices voor iSCSI. Een DGS-1224T kost (mpl.be) € 384.

The screenshot shows the web management interface for a D-Link DGS-1224T switch. The browser address bar shows 'http://192.168.36.32/'. The interface includes a navigation menu on the left with sections for Setup, Maintenance, and Logout. The main content area is divided into two sections: 'Switch Status' and 'PORT Status'.

Switch Status

Product Name	DGS-1224T
Firmware Version	3.00.17
Protocol Version	2.001.001
IP Address	192.168.36.32
Subnet mask	255.255.255.0
Default gateway	192.168.36.1
Trap IP	0.0.0.0
MAC Address	00-13-46-36-68-C6
System Name	Switch 1
Location Name	zwarte rack
Login Timeout (minutes)	5
System UpTime	93 days 7 hours 29 mins 52 seconds

PORT Status

ID	Speed		Flow Control		Default Priority	ID	Speed		Flow Control		Default Priority
	Set	Status	Set	Status			Set	Status	Set	Status	
10/100/1000 Mbps											
01	Auto	Down	Enable	Off	0	02	Auto	1G Full	Enable	On	0
03	Auto	Down	Enable	Off	0	04	Auto	100M Full	Enable	Off	0
05	Auto	1G Full	Enable	On	0	06	Auto	1G Full	Enable	On	0
07	Auto	10M Full	Enable	On	0	08	Auto	Down	Enable	Off	0
09	Auto	1G Full	Enable	On	0	10	Auto	1G Full	Enable	On	0
11	Auto	100M Full	Enable	Off	0	12	Auto	Down	Enable	Off	0
13	Auto	1G Full	Enable	On	0	14	Auto	1G Full	Enable	On	0
15	Auto	1G Full	Enable	On	0	16	Auto	1G Full	Enable	On	0
17	Auto	Down	Enable	Off	0	18	Auto	10M Full	Enable	On	0
19	Auto	Down	Enable	Off	0	20	Auto	Down	Enable	Off	0
21	Auto	100M Full	Enable	On	0	22	Auto	100M Half	Enable	On	0
23	Auto	1G Full	Enable	On	0	24	Auto	Down	Enable	Off	0

Figuur 91: De D-Link DGS-1224T, een layer-2 gigabit switch met 24 poorten.

Verder is er ook voor iSCSI een storage array nodig, echter hier zijn de opties legio. Er zijn 3 oplossingen (van goedkoop naar duur): een gewone server met een geïnstalleerde iSCSI

Target, een Storage Server met een geïnstalleerde iSCSI Target of een complete Storage Appliance.

- De eerste kan om het even welke server zijn, waar we dan een iSCSI Target op installeren. Deze kunnen we dan meteen inzetten als iSCSI Server als er maar genoeg harde schijven in zitten. Dit vereist wat kennis en werk, maar kost evenveel als een gewone server+extra schijven + iSCSI Target software.
- Een Storage Server is een speciale server die eigenlijk uit 2 delen bestaat, zoals de in ons lab geteste Intel SSR212MC2. Vooraan zit er een Storage Array waar een groot aantal standaard schijven (SAS of SATA) hun plaats vinden. Achter die Storage Array zit alles wat in een normale server ook zit, inclusief nog extra plaats voor 1 of meerdere systeem schijven. Deze kunnen we ook zelf volledig installeren en opzetten als iSCSI Target. Deze server is iets duurder dan een standaard server (\$2800 zonder harde schijven). Maar is wel wat veiliger en gebruiksvriendelijker.



Figuur 92: Intel SSR212MC2, een storage server

- De duurste oplossing: een Storage Appliance ziet er hetzelfde uit als een Fibre Channel storage array maar dan met vooraf geïnstalleerde iSCSI software erbij. Hij heeft een aantal netwerk-poorten om zo meteen ingeschakeld te worden op het netwerk. Via een management-interface kunnen we dan alle instellingen opzetten. Voor bijvoorbeeld een VTrak M500i (\$4750). Er is ook een iSCSI Storage Appliance oplossing van HP, die bestaat uit enerzijds de MSA1510i (een 2U iSCSI controller) en anderzijds een HP Storage Enclosure (bijv. een MSA20) die aan de controller moet gehangen worden. De Controller & Enclosure samen kosten \$5999, of de controller alleen kost \$3799.



Figuur 93: De Promise VTrak M500i, een storage appliance

Als we alles optellen komen we aan een prijs voor een switch plus een storage server op € 1798 + € 384 = **2173 euro**. Aanzienlijk goedkoper dan een Fibre Channel oplossing.

Let er ook op dat als de switch (of het nu een ethernet of fibre channel switch is) uitvalt, ook alle virtuele servers uitvallen. Het is misschien aangeraden om het aantal switchen & kabels te verdubbelen om zo voor redundancy te zorgen. Dit zou voor een nog groter prijsverschil zorgen tussen iSCSI en Fibre channel.

In deze berekeningen werd voorlopig geen rekening gehouden met de prijs van iSCSI Software. Dit komt doordat we dit kunnen opzetten zonder meerkost als we willen. In SUSE Linux Enterprise Server 10 zit namelijk standaard een iSCSI Target (iSCSI Enterprise Target). En zoals bekend is dit alles gratis. Tevens zijn de prestaties van deze target vergelijkbaar met die van een gelijkaardig Windows platform.

Wat de client-versie (iSCSI Initiator) betreft, ook deze is in alle gevallen gratis. De meest gebruikt is bijvoorbeeld de Microsoft iSCSI Initiator 2.05 die gratis af te halen is van de Microsoft website.

[Tijl Deneut – Student Master Electronica-ICT]

9.2. Case Study 2: Savaco High Availability solutions

CASESTUDY SAVACO P2V

www.savaco.be – info@savaco.be – President Kennedypark 24, B-8500 Kortrijk

Tel: +32/(0)56 26 03 61

VOORSTELLING BEDRIJF



Savaco werd opgericht op 4 februari 1991 door Carl Sabbe en Rik Vandemoortele en is sinds 1995 gevestigd in het Kennedypark te Kortrijk. Het gaat ze vlug voor de wind en is trots de enige Belgische partner van HP te zijn. Savaco wordt op IT-vlak door Microsoft erkend als Certified Solution Provider. Vanaf 2005 is Savaco zelfs een Microsoft Gold Partner.

Savaco heeft een 40-tal werknemers in dienst, verdeeld over sales & marketing, techniek & ontwikkeling, operations en administratie.

ACTIVITEITEN BEDRIJF

SAVACO levert toegevoegde waarde en informaticaoplossingen in het domein van informatietechnologie en engineering. Productiviteit, efficiëntie en langetermijnrelaties staan hierbij centraal. Dit wordt bevestigd door meer dan 400 bedrijven die de vakkennis en expertise van SAVACO gebruiken als ondersteuning voor het realiseren van kortere time-to-market, reductie van kosten, innovatie en communicatie. De fundamenten voor dit succes zijn gebaseerd op de integratie van "Best-in-Class"-oplossingen.

Savaco heeft 2 grote onderdelen, **Networking** en **Engineering**.

Engineering omhelst het ontwerpen van praktische systeemoplossingen, met gespecialiseerde CAD-tools. Savaco kan alles verzorgen: opleiding, helpdesk, on-site support, integratie, data-uitwisseling en implementatie.

Networking omhelst het opzetten van een volledig bedrijfsnetwerk. Ook hier komen alle onderdelen aan bod die daarmee kunnen te maken hebben: aankoop en installatie hardware en bijhorende software, diensten, opleidingen, beveiliging, advies en support.

VOORSTELLING CASE

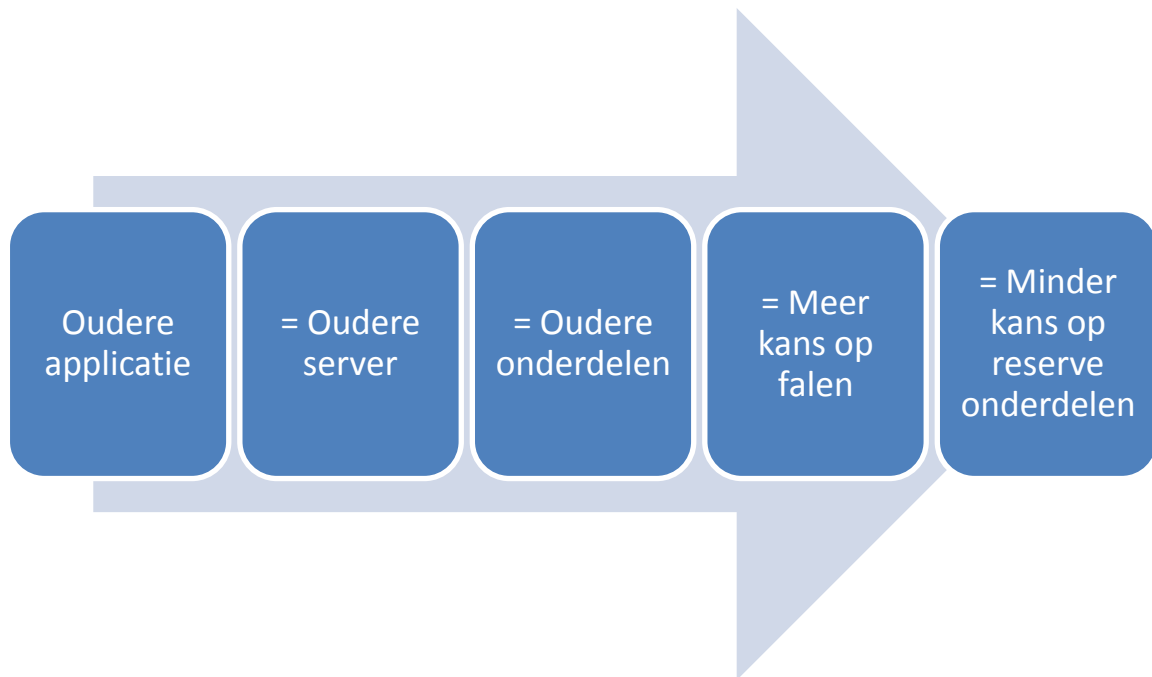
PROBLEEMVOORSTELLING

Onder hun netwerktoepassingen willen zij graag high availability nastreven. Ze willen hun klanten verzekeren dat hun applicatie zo snel mogelijk terug online is na het falen van bepaalde hardware.

Een applicatie die continu beschikbaar moet zijn, draait meestal al een hele tijd (misschien jaren) op dezelfde machine. Doordat de applicatie al lang draait, kan ook de server ouder zijn. Dit betekent meestal oudere onderdelen en het vervangen van deze oudere onderdelen is niet altijd simpel.

Als deze server dan uitvalt, wordt het soms zeer moeilijk of onmogelijk om reserve onderdelen te vinden die nog compatibel zijn met deze machine. Bijgevolg ligt de machine uren/dagen/weken stil, omdat men de applicatie moet ombouwen naar een nieuwe machine, of omdat men dat ene onderdeel nog moet zien te vinden. Dit is een groot en zeer reëel probleem waar Savaco vaak mee wordt geconfronteerd.

Dit komt de zogenaamde *Business Continuity* niet ten goede en dat is één van de doelstellingen die Savaco wil aanleveren aan haar klanten.



OPLOSSINGSVOORSTELLING

De typeoplossing die Savaco voorstelt is een zogenaamde P2V, wat staat voor Physical-to-Virtual.

Met deze nieuwe technologie wordt gebruik gemaakt van virtuele machines. Als partner van VMWare, en doordat ze enkele VCPs (VMware Certified Professional) in hun rangen hebben, hebben ze zeker genoeg kennis van virtuele machines om een virtueel machinepark aan te bieden.

P2V is een manier om snel, goedkoop en vooral stabiel een bestaande fysieke machine om te zetten naar een of ander virtueel platform. Het volledige besturingssysteem wordt omgezet en kan meteen opstarten om het fysieke platform te kunnen overnemen of ontlasten.

P2V is een recente en nieuwe technologie, en er is nog maar zeer weinig kennis hieromtrent. Dan kan de volgende vraag rijzen: is P2V wel snel en vooral stabiel genoeg? Wat is de kans dat P2V werkt en werkt het in alle omstandigheden.

Savaco werkt met een zeer brede variëteit aan klanten en hun applicaties zijn al net zo

variabel zoals domeincontrollers, netwerken, databases die werken op DNS namen, NETBios ... of zeer specifieke besturingssystemen; # Windows systemen, # Linux systemen en soms BeOS wordt gebruikt. Is dit allemaal even stabiel om te zetten? Welke garantie kan Savaco op dit gebied leveren aan hun klanten (welke prijssetting kunnen ze hanteren)?

P2V, CONCREET

Hoe gaat het omzetten van een fysieke machine naar een virtuele nu concreet?

Er bestaan 2 soorten omzettingen, de zogenaamde offline of online conversies. Bij VMware heten deze respectievelijk de cold of hot cloning. Dit slaat op de fysieke machine.

Bij een offline conversie moet de machine worden afgezet om dan gestart te worden in een andere omgeving (meestal met een boot-cd). Van die omgeving wordt de harde schijf (of schijven) ingelezen en overgezet. Het voordeel is dat er geen footprint achterblijft op de fysieke machine, er wordt niks op geïnstalleerd en soms heeft deze zelfs niet door dat er een conversie is gebeurd.

Bij een online conversie wordt er een applicatie geïnstalleerd op het fysieke systeem en leest deze alles in. Het voordeel is dat er weinig hardware afhankelijkheden zijn, er is geen boot-cd die bijvoorbeeld specifieke opslag drivers moet bevatten om de schijven te kunnen lezen.

Waar zitten nu de moeilijkheden bij het uitvoeren van een P2V?

Het eerste deel is redelijk simpel, gewoon de harde schijf (of schijven) byte per byte over kopiëren naar de virtuele harde schijf. Alle bestanden worden daarbij gewoon overgezet, en het bestandssysteem wordt behouden.

Het tweede deel dient om het besturingssysteem daadwerkelijk te kunnen opstarten in de nieuwe, virtuele omgeving. Uiteraard zijn de meeste drivers niet meer nodig en worden ze door het P2V-programma overboord gesmeten. Dit werkt uiteraard volledig anders onder Windows dan onder Linux of ander besturingssystemen. Nadat de overbodige drivers eruit werden gehaald moeten de juiste, nieuwe drivers geïnjecteerd worden in de harde schijf image. Deze zijn verschillend per virtueel platform. Soms moet ook de HAL aangepast worden. De CPU die virtuele machines van VMware zien is een Intel Genuine of AMD Opteron CPUs. Terwijl sommige fysieke machines dan weer een AMD CPU aan boord hebben. Hetzelfde geldt voor de chipset.

Doordat de meeste virtualisatie platformen specifieke tools hebben (VMware tools of Hyper-V Integrated Components), kunnen hieruit de drivers gehaald worden. Deze kan een P2V tool dan meteen injecteren in een harde schijf, uiteraard moet fysieke toegang dan mogelijk zijn tijdens de gehele conversie.

CONCRETE OPLOSSINGSMOGELIJKHEDEN:

-> MARKTOVERZICHT

P2V oplossingen zijn dun bezaaid, maar de meeste, zo niet allen, zijn gratis.

- UltimateP2V (gratis);
gecreëerd door Qui Hong, Chris Huss & Mike Laverick. Het maakt gebruik van “BartPE” (Bart Preinstalled Environment) om in het fysieke systeem én het virtuele systeem op te starten. Het gebruikt dan Symantecs Ghost Software om de harde schijf te klonen.
- VMware P2V assistant (\$ 2500 voor Starter & \$ 6100 voor Enterprise);
dit was de oudere versie van VMware converter en wordt niet meer gebruikt. Het was vooral aangeraden door VMware zelf, indien er slechts enkele conversies op het programma stonden. Ik bespreek dit niet verder.
http://www.vmware.com/support/p2v21/doc/p2v_manual_webTOC.html Het product is end-of-life en officiële support eindigt op 8 Augustus 2008.
- VMware converter 3.0.2 (gratis voor standaard versie);
is de meest bekende oplossing en is misschien voor Savaco de meeste logische oplossing.
- Microsoft Virtual Server 2005 Migration Toolkit (VSMT, gratis);
werkt enkel samen met Virtual Server 2005, dus na deze P2V zou een nieuwe V2V nodig zijn. Dit is omslachtig en bovendien worden enkel een 4-tal Windows besturingssystemen ondersteund. Met de komst van Hyper-V op Windows Server 2008 zal deze oplossing wellicht verdwijnen. Om die redenen wordt dit niet verder besproken.
<http://www.microsoft.com/technet/virtualserver/downloads/vsmt.mspx>
- PlateSpin powerconvert (\$ 200 per gelukte overzetting);
dit is een totaaloplossing die zich wil specialiseren in alle mogelijk conversies. Ze bieden Disaster Recovery Planning en zware on-site support of via resellers. Er is een “Live Transfer” optie om een draaiende fysieke machine on-the-fly over te zetten, maar dit werkt enkel bij Windows Systemen. Ze hebben ook grote problemen om volop gebruik te maken van het netwerk, er worden zelden snelheden boven 400KBps gerapporteerd. Slechts enkele Linux-versies worden ondersteund en het grootste gemis is dat er op geen enkel gebied ondersteuning is voor 64bit besturingssystemen. Er is ook geen trial te vinden, dus dieper kunnen we hier niet op ingaan.
- Leostream P2V direct 3.0 (\$ 100 per gelukte overzetting);
is een iets geavanceerdere manier, het is een volledig platform. Maar de officiële productsite ligt reeds 3 weken plat, alsook de trial download-pagina. Waardoor er

een vermoeden is dat deze oplossing ter ziele is gegaan...

<http://www.leostream.com/signUpForm.html>

- Acronis True Image Echo Server 9.1(\$ 49,99 per licentie); vanaf versie 9.1 is True Image in staat om een ghost-image van een bestaand systeem te nemen en deze om te zetten naar een ander platform, inclusief VMware virtuele schijven.
- Vizioncore vConverter (\$ 179 per licentie per jaar); is partner van VMware en werkt bijgevolg enkel met VMware Infrastructure 3. Het is eigenlijk opgezet als backup oplossing voor bestaande fysieke systemen. Het voert een P2V uit terwijl de fysieke machine draait. Eén van hun doelstellingen is "P2V Disaster Recovery" waardoor dit zeker een kandidaat is.

DETAILOVERZICHT VAN ENKELE P2V PROGRAMMA'S

ULTIMATE P2V

Dit is eigenlijk een plugin voor BartPE die zorgt voor de aanpassing van de HAL en andere drivers. Het zorgt eigenlijk voor een ghost van de fysieke machine en gaat daarop de aanpassingen doorvoeren. Het ghost programma zelf heeft dus ook een plugin nodig voor de BartPE boot-cd. We hebben getest met Symantec Ghost 8, omdat deze zeker ondersteund wordt. **Het grote voordeel van deze manier is de betere ondersteuning voor Linux besturingssystemen.**

Een typisch stappenplan zou er als volgt uitzien:

- Maak een UltimateP2V opstartbaar CD/iso bestand.
- Creëer een virtuele machine
- Start zowel virtuele als fysieke machine met de opstartbare UltimateP2V CD/iso
- Kloon de harde schijf van de fysieke machine mbv Symantec Ghost.
- Start de virtuele machine en configureer het besturingssysteem.

Nadelen zijn dat de boot-cd de netwerk-drivers nodig heeft van zowel de fysieke als de virtuele machine. Voor VMware is dit vooral de VMXnet driver. Ook de LSI Logic & BusLogic SCSI-drivers dienen geïmporteerd te worden in de boot-cd, omdat deze de meest gebruikte drivers zijn onder Windows & VMware virtualisatie.

Grootste nadelen:

Dit lijkt een zelf-hulp oplossing te zijn, waar veel trial-and-error bij komt kijken. 1 cd maken met alle drivers die fysieke machines kunnen hebben, is onmogelijk.

Het blijft tevens koffiedik kijken in hoeverre de plugin (geschreven door Qui Hong) werkt om de HAL aan te passen en de drivers om te wisselen. Er zijn reeds gevallen bekend waar een IDE-schijf niet om te zetten was naar een virtuele SCSI-schijf.

Grootste voordelen:

Omdat het een gratis (open-source) programma is, is er veel informatie te vinden op internet.

Het lijkt perfect Windows en Linux door elkaar te ondersteunen en zou in theorie ook samenwerken met andere virtualisatie-platformen als Hyper-V en Xen.

Wie geen werk schuwt kan hier zeer mooie dingen mee doen.

In het geval van Savaco; een P2V heeft net evenveel kans op slagen als een V2P en werkt op exact dezelfde manier, extra kennis is niet nodig.

http://www.rtfm-ed.co.uk/?page_id=174

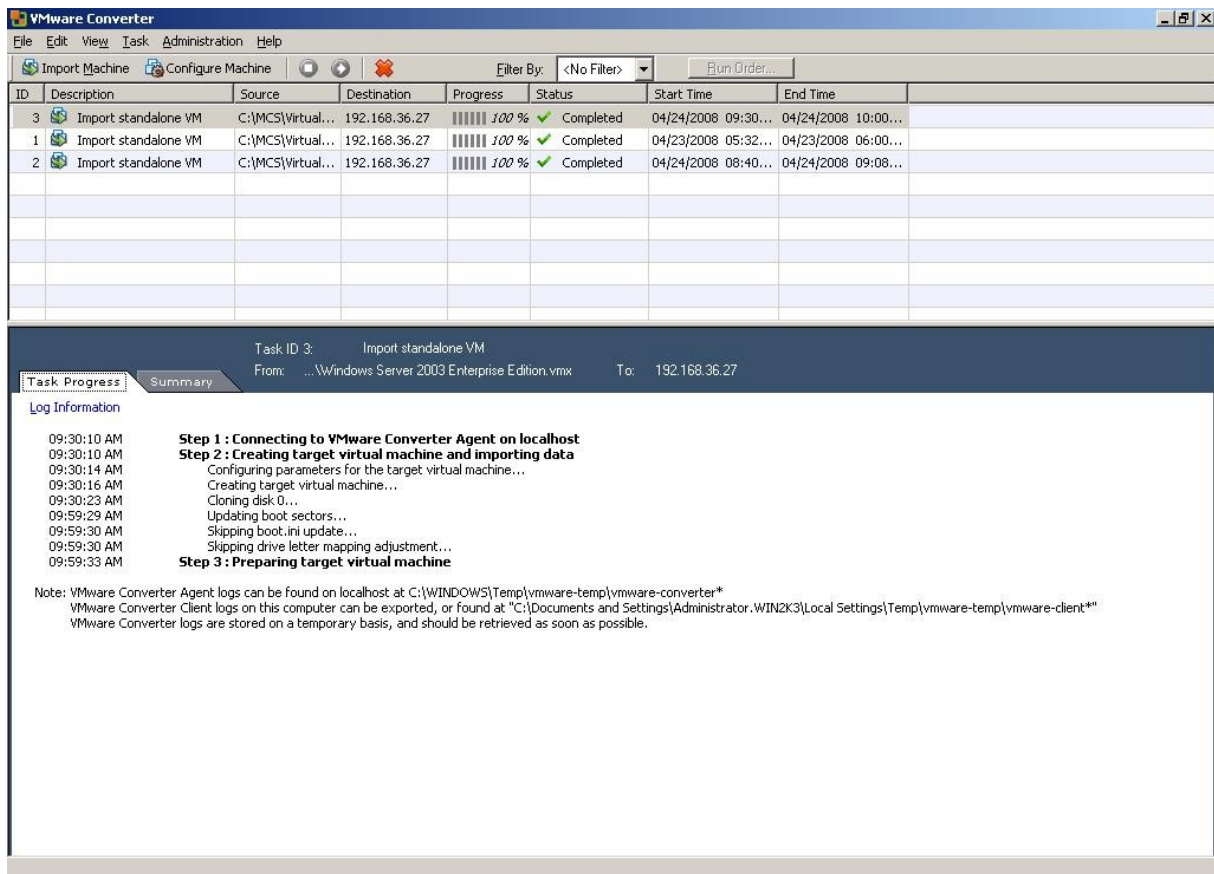
VMWARE CONVERTER 3.0.2

Met VMWare converter zijn we reeds het meeste bekend, omdat . Het is geschreven door VMware en kent een zeer grote slaagkans bij het overzetten van fysieke naar VMware machines. Zowel ESX, GSX, Server als Workstation worden ondersteund en ook omzetten tussen deze VMware producten onderling wordt ondersteund.

Hoewel officieel Linux niet ondersteund wordt, zijn er reeds gevallen van succesvolle conversies bekend. Echter enkel als de fysieke machine werkt met SCSI schijven, en enkel cold cloning (offline conversie) is mogelijk.

Er is een Starter en een Enterprise versie. De Starter is gratis terwijl de Enterprise wordt meegeleverd met het VirtualCenter Management Server product.

Het verschil is dat bij de Starter versie je enkel een hot cloning (online conversie) kan doen, terwijl de Enterprise via een boot-cd kan werken. Alsook meerdere conversies op hetzelfde moment zijn enkel met de Enterprise versie mogelijk.



Figuur 94: Een aantrekkelijk GUI zorgt ervoor dat VMware gebruikers zich meteen thuis voelen.

Grootste nadelen:

Door VMware ontworpen en daardoor enkel support onder de verschillende VMware producten.

Geen support voor Linux.

Grootste voordelen:

Meteen installatie van de VMware Tools tijdens de conversie.

Grote kans op slagen bij Windows systemen, tenzij bijvoorbeeld de harde schijven geïncrypteerd zijn zodat de converter niet aan de bestanden kan.

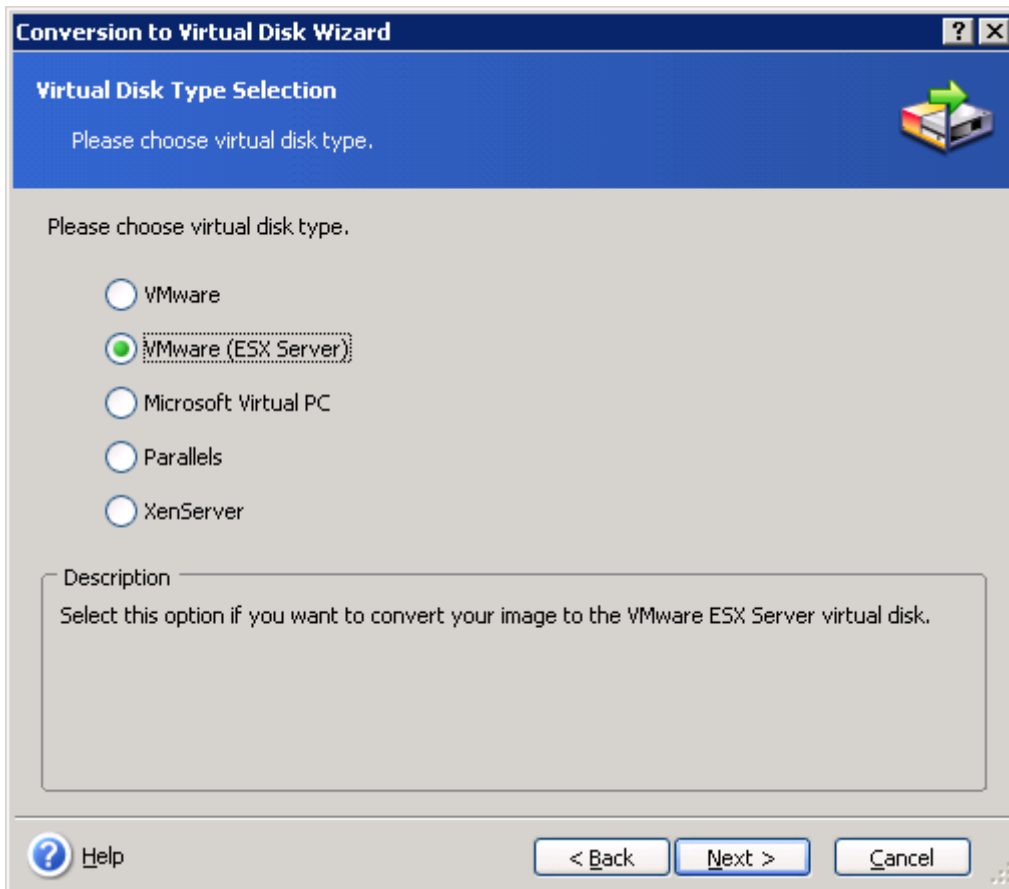
Ze hebben tevens een boot-disk, zodat een footprint op het fysieke systeem niet noodzakelijk is.

Veel opties. Tijdens het converteren is er de mogelijkheid om de hostname en netwerkadressen aan te passen, specifieke SIDs toekennen, ... Zelfs syspreppen is een optie.

<http://www.vmware.com/products/converter>

ACRONIS TRUE IMAGE ECHO SERVER 9.1

Acronis heeft vanaf versie 9.1 in zijn server software de optie voorzien om volledige schijf images om te zetten naar virtuele schijven.



Figuur 95: Eenvoudige conversie wizard van Acronis

Het maakt een plat systeem van uw schijf. Als uw originele schijf 40GB groot is, maar er is maar 1GB gebruikt (wat resulteert in een .tib image-bestand van 1GB) dan zal het vmdk bestand toch 40GB groot zijn. Alsook de HAL wordt aangepast en bepaalde drivers geïnjecteerd om het OS te laten opstarten in de virtuele omgeving. Daarna is het gewoon een kwestie om de (soms hele grote) vmdk-bestanden te kopiëren naar de ESX server en deze als virtuele schijven aan een (nieuwe) virtuele machine te hangen.

Als we voor disaster recovery zouden kiezen voor Acronis TrueImage dan hebben we als voordeel dat we makkelijk en snel images kunnen maken van bestaande, draaiende fysieke systemen. En deze dan opzij zetten, eventueel met incrementele images bijgewerkt. Dit bespaart op ruimte en als er zich een ramp voordoet kunnen we desbetreffende images overzetten naar een virtueel platform en verder gebruiken.

Dit bespaart op schijf- en machineruimte en kan een snelle tussentijdse oplossing zijn.

Grootste nadelen:

Altijd migratie via een tussenstap (Physical naar Ghost Image naar Virtual Image naar Virtual

server)

Geen support voor de conversies, een test op voorhand is altijd aangeraden

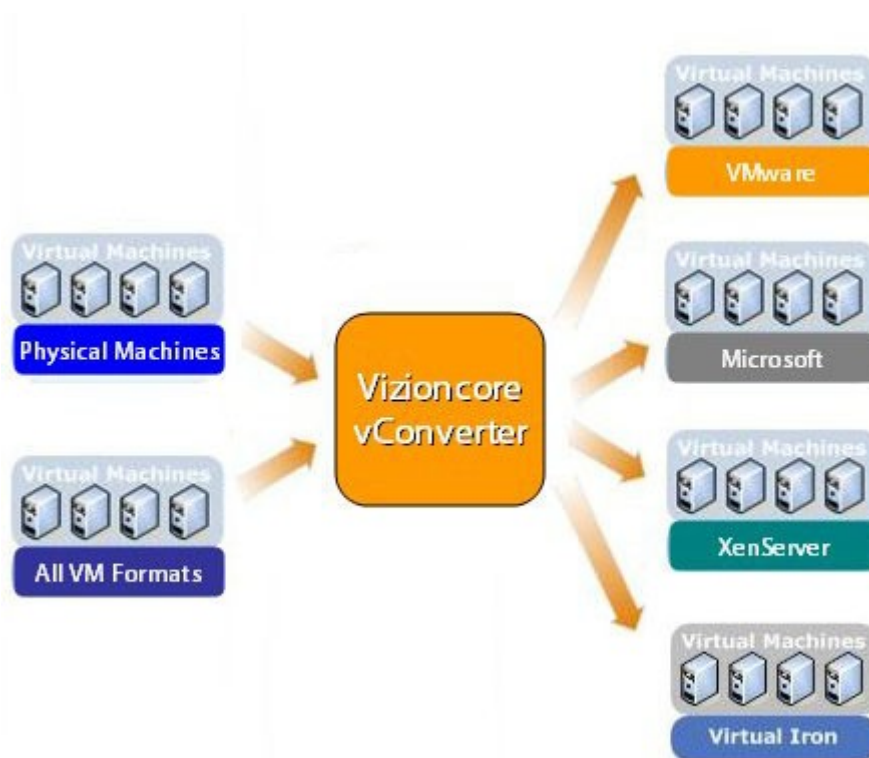
Grootste voordelen:

Goed idee voor als er reeds image-backups bestaan, makkelijk te implementeren in een bestaand backup-protocol.

Mogelijkheden tot converteren naar de virtuele schijf-images van VMware, de VMware ESX Servers, MS Virtual PC (incl Virtual Server 2005), Parallels en XenServer.

<http://www.acronis.com/pr/2007/01/pr01-24-p2v-v2v-server-migration.html>

VIZIONCORE vCONVERTER



Figuur 96: Omzetten van alle fysische & virtuele formaten naar alle virtuele formaten.

Vizioncore vConverter is een totaaloplossing, ze bieden zelfs support voor disaster recovery plannen.

Grootste nadelen:

Het is niet gratis, waardoor het jaarlijks en per licentie een behoorlijke kost is. Daardoor is de drempel behoorlijk groot.

Voorlopig is er nog geen grote userbase, dus weinig ervaring te vinden op internet. Maar er is wel veel reclame te vinden.

Ondersteuning voor enkel Windows 2000, Windows 2003 & Windows XP.

Er is een derde machine nodig, waar de software op draait.
Het is redelijk nieuw (release finale versie: Februari 2008) en dus wellicht niet bugvrij.

Grootste voordelen:

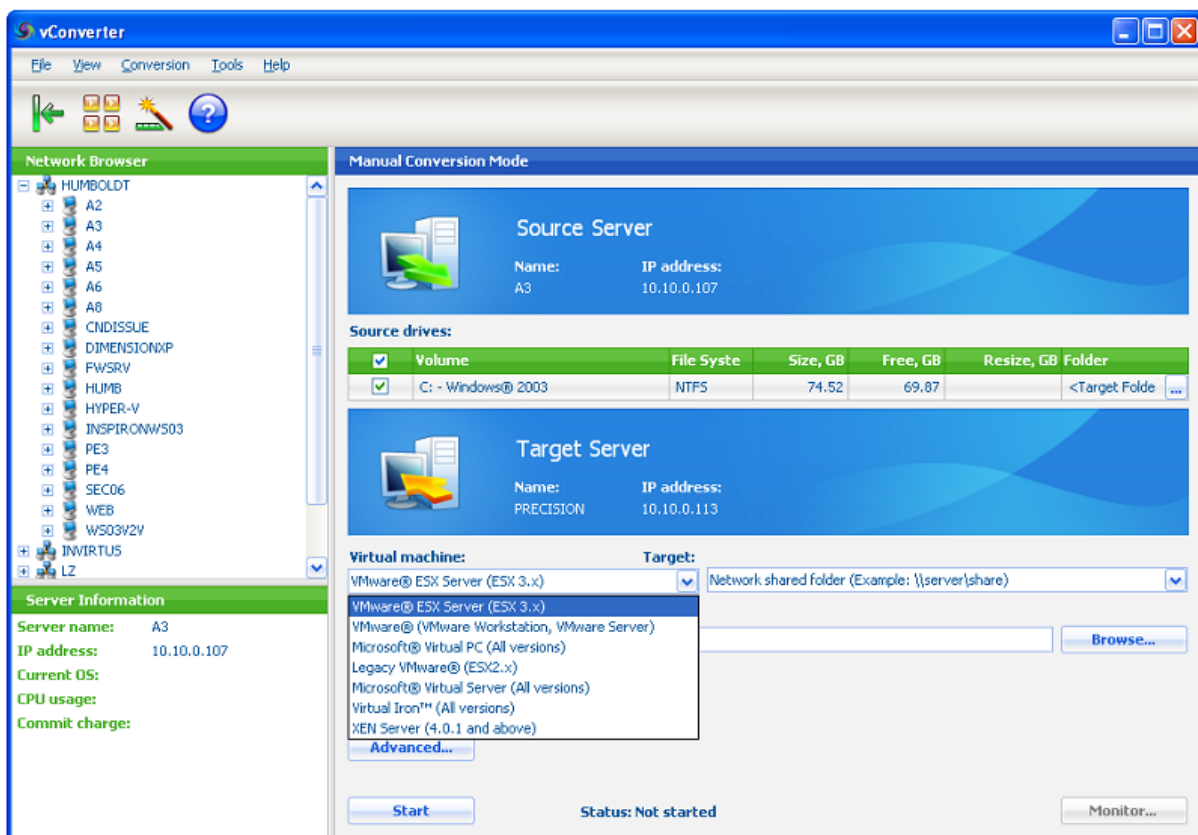
Mooie, eenvoudig GUI en ze gaan prat op een zeer snelle conversie.

Voor de prijs krijg je een goeie support, met garanties voor conversies.

Automatisatie mogelijk via een Command Line Interface.

Werkt samen met alle virtualisatie platformen, en geoptimaliseerd voor ESX Server.

Geen reboot nodig voor en geen footprint op de fysieke machine.



Figuur 97: Intuïtieve GUI van vConverter

<http://www.vizioncore.com/vConverter.html>

BESLUIT

De keuze zal van verschillende factoren afhangen, er is geen perfecte oplossing voor alle mogelijkheden.

De eerste factor is het virtueel platform, voor een VMware product is het best om VMware converter te gebruiken. Hyper-V of Xen gebruikers hebben het meeste baat bij UltimateP2V of Vizioncore vConverter.

De tweede factor is het besturingssysteem zelf. Meestal wordt Windows wel zonder problemen ondersteund, maar is het koffiedik kijken voor Linux systemen.

Het voorstel zou kunnen zijn om alles eerst te proberen met VMware converter. Als het niet lukt met deze converter, of er is een ander platform nodig dan VMware, dan is er altijd de mogelijkheid om te kiezen voor UltimateP2V. Aangezien beide producten geen installatie nodig hebben op de fysieke machine is er weinig of geen risico.

Een derde factor is die kritiekheid van de fysieke machine. Als deze op geen enkel moment mag afgezet worden, is een installatie van VMware converter (op een dood moment van de week) een van de weinige opties.

[Tijl Deneut – Student Master Electronica-ICT]

LITERATUURLIJST

- [1] <http://www.michaeljordan.nl/SAS.html>
- [2] <http://tweakers.net/specials/server-storage/article/6/storage-is-meer-dan-opslag.html>
- [3] <http://blogs.msdn.com/larryosterman/archive/2005/08/31/458572.aspx>
- [4] <http://www.microsoft.com>
- [5] <http://www.iometer.org/>
- [6] <http://www.microsoft.com/Downloads/details.aspx?familyid=9A8B005B-84E4-4F24-8D65-CB53442D9E19&displaylang=en>
- [7] <http://en.wikipedia.org/wiki/Xen>
- [8] <http://etbe.coker.com.au/2007/01/01/installing-xen-domu-on-debian-etch/>
- [9] http://wiki.kartbuilding.net/index.php/Create_DomU
- [10] <http://en.wikipedia.org/wiki/Lvm>
- [11] W. von Hagen. Xen Virtualization. Wiley Publishing, Inc. {978-0-470-13811-3}
- [12] Thesissen van voorgaande jaren: Sizing Servers Lab: Pieter Beel & Stijn Verslyckens
- [13] Virtualization for Dummies, AMD Special Edition {978-0-470-13156-5}

APPENDICES

Appendix A: SATA/SAS bekabelings Overzicht

Algemeen

Ik schreef dit artikel zodat iedereen een klein overzicht heeft van de mogelijke SAS-bekabelingen en hun benamingen. Op de interne website van het lab wordt dit voorgesteld als een tabel, maar om praktische redenen houden we het hier gewoon op een opsomming.

Intern SAS, IPASS to IPASS

Ook bekend onder de naam Multilane Cable



Figuur 98: SFF-8087 Multilane Cable

Extern miniSAS 8470

Ook bekend als een miniSAS-36 aansluiting



Figuur 99: SFF-8470 extern miniSAS protocol

Extern miniSAS 8088

Ook bekend als een miniSAS-24 aansluiting



Figuur 100: SFF-8088 extern miniSAS protocol

Extern Infiniband protocol voor gebruik met Enclosure



Figuur 101: SFF-8470 (Enclosure) naar SFF-8088 (Server)

Interne SAS voor gebruik van backplane naar controller



Figuur 102: SFF-8484 (backplane) naar SFF-8087 (controller)

Interne SAS/SATA voor gebruik van apparaten naar backplane



Figuur 103: SFF-8484 (backplane) naar SFF-8482 (apparaat)

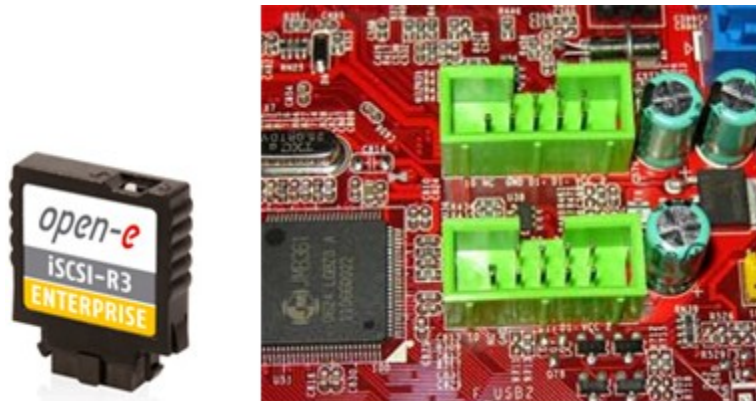
Appendix B: Eerste ervaringen met Open-e

Inleiding

Dit is een iSCSI (maar ook NAS) oplossing die zeer dicht op de hardware zit.

Het komt erop neer dat het een iSCSI Target is die in een eigen OS draait (unix-based) en die kan opgestart worden van een CD.

Maar voor performantie redenen hebben ze ook USB-module's die rechtstreeks op het moederbord geprikt worden (zie foto's hieronder).



Figuur 104: Open-e iSCSI Target Module, voor op een USB-poort zoals rechts afgebeeld.

Eerste ervaringen, concrete showcase

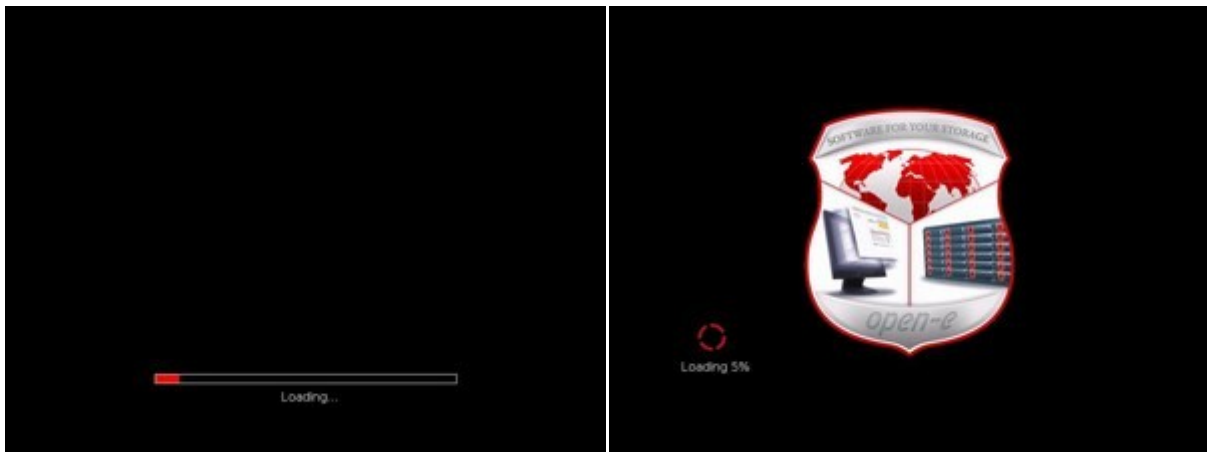
De CD heet `demo_10.n20070213s.210.b2573.iscsir3-ent.oe.iso` en komt met een pdf die makkelijk uitlegt hoe Open-e te gebruiken is. Hij werd hier voor het eerst getest in een virtuele omgeving, toch werkt alles naar behoren.

Requirements:

- x86 - compatible PC
- 1 GByte RAM
- CPU: 2,8GHz
- Ethernet Card
- Onboard USB Connector

Deze requirements spreken voor zich en zijn niet overdreven, uiteraard geldt dat een beter systeem wellicht aangeraden is.

1. Opstarten van CD



Figuur 105: Open-e: 2 laadschermen

```
Welcome to Open-E (1921-83) (Press F1 for help)
-----
Model:      Open-E (1921-83)
Version:    2.10.1100000000.2573
Release date: 2007-02-13
S-M:       1357106427

Network settings:
Interface 1: eth0 IP:192.168.0.220/255.255.255.0

NFS settings:
port:      443
allow from: all

This is TRIM version, 30 days left for evaluation.

Selftest OK.
```

Figuur 106: Open-e het startscherm met een IP-adres melding

Uiteindelijk komen we in dit scherm, het ip staat standaard op 192.168.0.220, dit is meestal niet goed, daarom verzetten we het. Dit doen we met de toetsencombinatie *<Ctrl>+<Left Alt>+<N>*.

We kiezen onze netwerkcontroller, drukken *<Enter>* en kiezen hier voor het gemak DHCP (met de pijltjes), opnieuw *<Enter>* en dan *<Spatie>* om DHCP te selecteren, terug *<Enter>*, dan Apply en het netwerk is ingesteld.



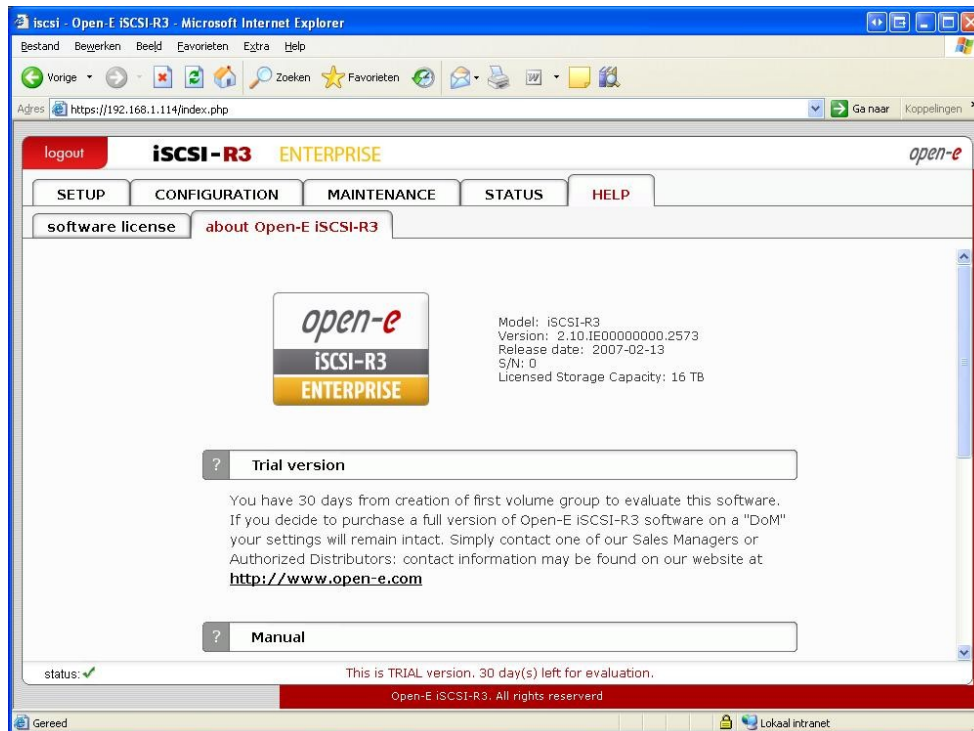
Figuur 107: Via de lokale console kunnen we het Open-e IP-adres instellen

We kunnen bepaalde instellingen controleren (oa tijd) met combinatie `<Ctrl>+<Left Alt>+<T>` maar laten dit nu achterwege.

2. Instellingen via Web-browser: Volume's aanmaken

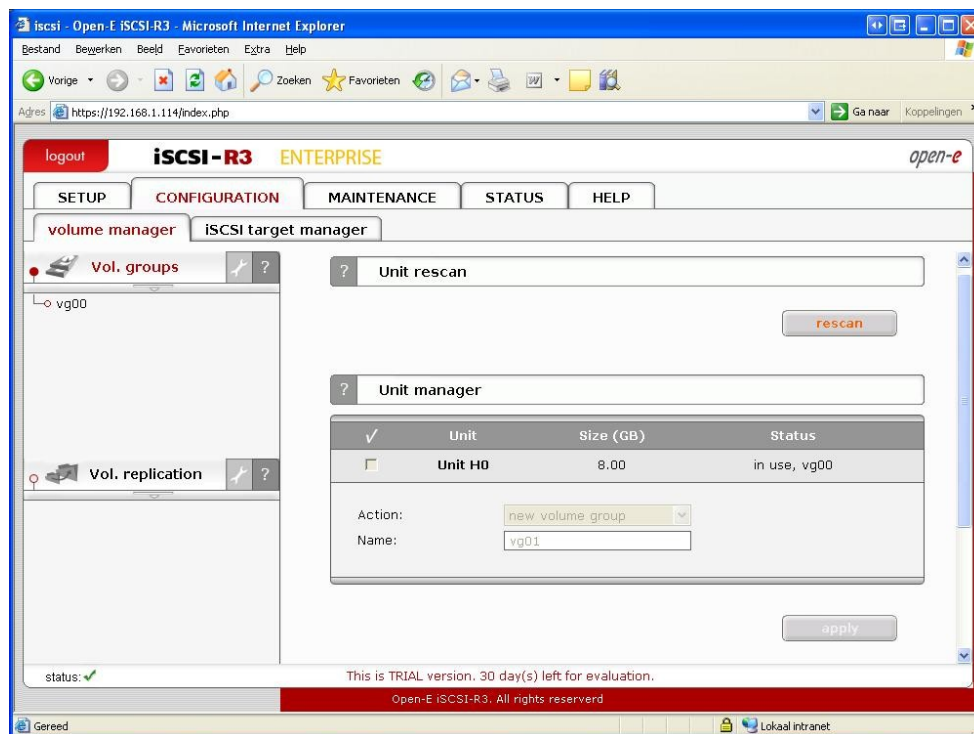
We gaan terug naar het eerste venster met Exit en laten de server voor wat hij is, we zoeken een client die op het netwerk zit en doen onze browser open, we tikken het eerder ingestelde ip in voor de configuratie, we merken ook meteen op dat het een SSL-verbinding betreft en kunnen dit alleen maar toejuichen.

Inloggen kan met het standaardwachtwoord *admin*, zonder hoofdletters, we komen op deze pagina:



Figuur 108: Open-e startvenster na inloggen via een webbrowser

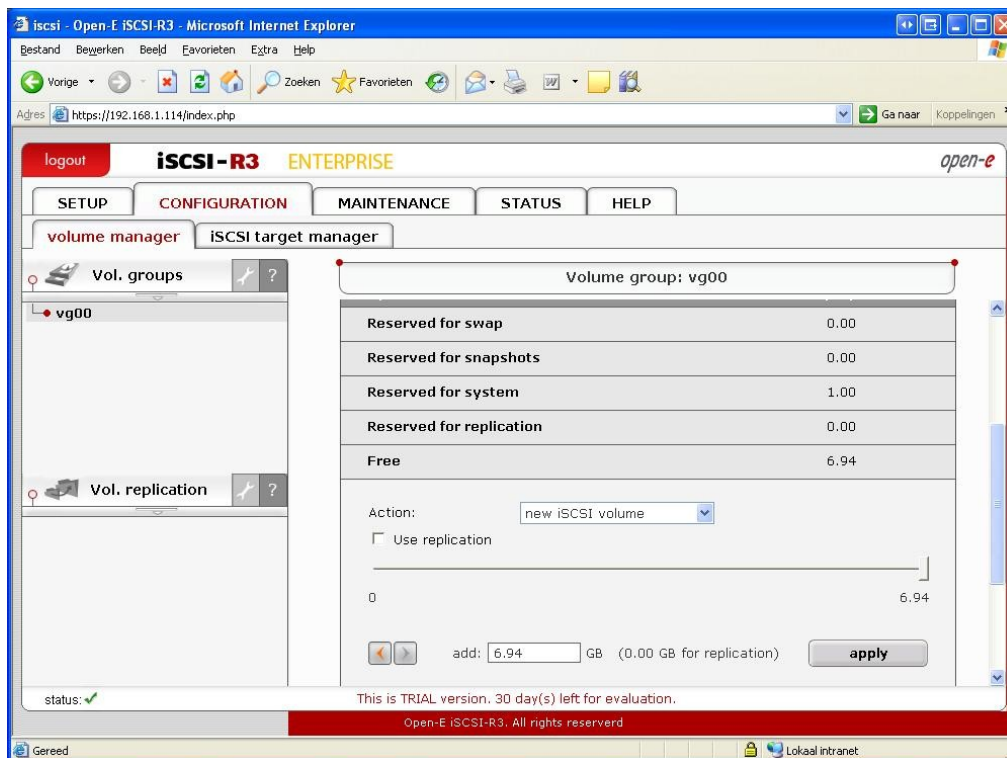
We maken eerst een volume aan, we gaan naar *Configuration* en dan *Volume Manager*; we vinken de eerste unit aan, we hebben maar 1 schijf en klikken op 'Apply', de schijf wordt dan een logisch volume, de eenheid waar iSCSI-R3 mee werkt.



Figuur 109: Open-e volume manager

Nu klikken we in het linkerdeel van het scherm op de naam van het logische volume, in ons geval 'vg00'. iSCSI-R3 heeft 1GB nodig voor zijn systeem en een SWAP van 4GB, daar dit

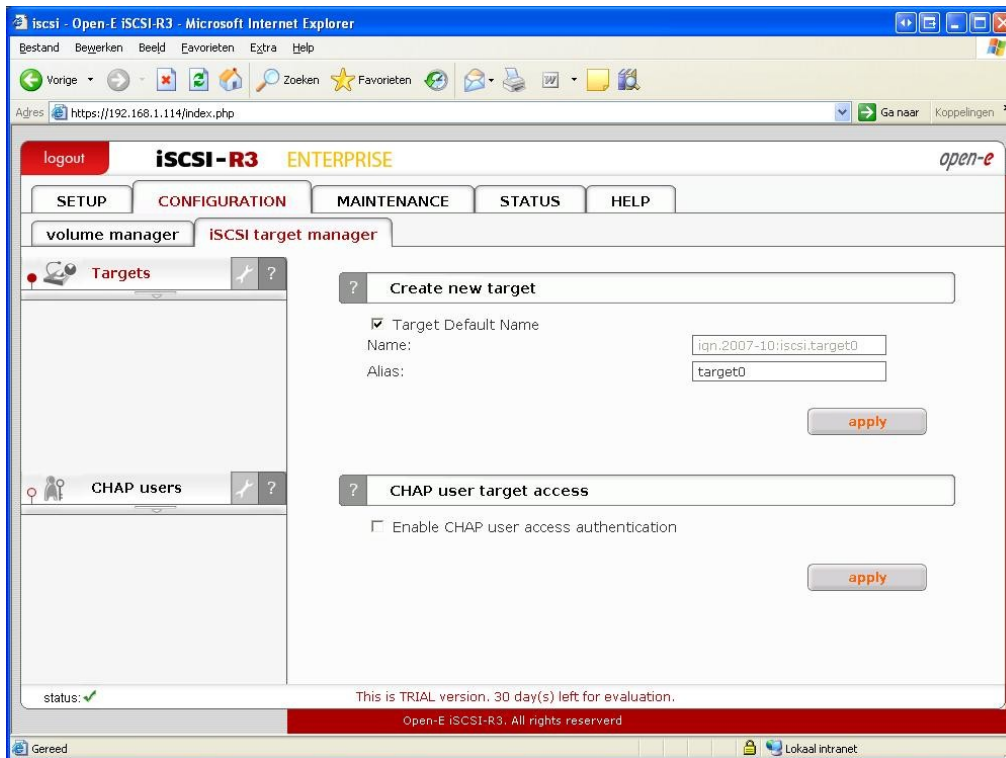
voor een test is gaan we de swap verwijderen, zo hebben we meer over voor onze partitie. We gebruiken de maximaal beschikbare ruimte door de schuifbalk naar rechts te verplaatsen en klikken op 'apply'.



Figuur 110: Open-e: Volume groups

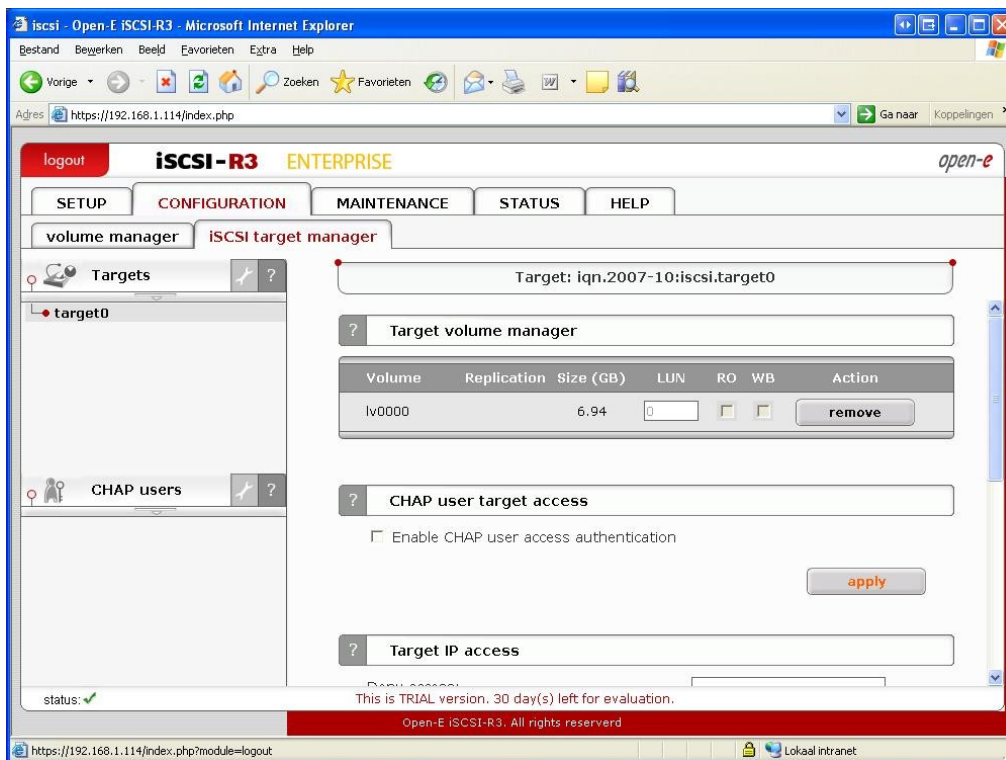
3. Instellingen via Web-browser: iSCSI Target's aanmaken

Nu klikken we op *Configuration*, gevolgd door *iSCSI target manager*, we kiezen een Alias en dan op *apply*.



Figuur 111: Open-e: aanmaken van een iSCSI Target

Daarna kunnen we opnieuw in het linkerdeel van het scherm op de naam van de target klikken en een volume toewijzen met de knop *add*.



Figuur 112: Open-e: volume toewijzen aan een target

Alles is opgemaakt en vanuit een client met een iSCSI Initiator (zoals MS iSCSI Initiator) kan er nu probleemloos verbonden worden met deze target.

Appendix C: Enkele voorbeelden van Storage Appliances

Algemeen

Kleine bespreking van de iSCSI servers die als iSCSI oplossing bestempeld worden en dit door verschillende spelers op de markt (Promise, HP, Intel ...)

Eerst het iets oudere product van Promise:

VTrak 15200

Zij hebben wel degelijk een iSCSI oplossing, namelijk de VTrak 15200 External Raid, productpagina informatie:



Figuur 113: Promise VTrak 15200 Storage Appliance

- 2x Gigabit Ethernet
- 1x Serieel
- 1x Ethernet Management
- Max 512MB Ram
- RAID 0, 1, 3, 5, 10 en 50
- SATA-150 (géén SATAII of SAS)
- 15x 3,5"
- 2x Redundant PSU

Prijs: \$4600 voor 256MB Ram versie zonder schijven

Er is een iets nieuwere versie van dit product:

VTrak M500i

Een nieuwe oplossing is de VTrak M500i; informatie:



Figuur 114: Promise VTrak M500i Storage Appliance

- 2x Gigabit Ethernet
- 1x Ethernet Management
- Max 512MB Ram
- RAID 0, 1, 1E, 5, 10 en 50
- SATAI & SATAII (geen SAS)
- 15x 3,5"
- 2x Redundant PSU

Prijs: \$4750 voor 256MB RAM versie

Ook HP heeft een iSCSI oplossing in hun productengamma:

HP MSA1510i

Ook HP heeft een iSCSI oplossing in hun aanbod, namelijk de HP StorageWorks 1510i Modular Smart Array. Deze server is een modulaair gebeuren, wat impliceert dat er een (HP) storage rack aan dient gehangen te worden, dit kan een MSA30 zijn (SCSI) of een MSA20 (SATA). De MSA1510i is de zogenaamde controller shelf die 2U hoog is.



Figuur 115: HP MSA1510i iSCSI Controller

- 2x Gigabit Ethernet (= 1 Ethernet iSCSI Module)
- Max 512MB Ram
- RAID 0, 1, 01, 5 en 6
- SATAII OF SCSI
- Afhankelijk van het aantal Enclosure's (tot 8 stuks ondersteund) kun je tot 16TB gaan (SCSI) of 48TB (SATA).
- 2x Redundant PSU

Prijs: \$3799 voor Controller Enclosure alleen (MSA1510i), \$5999 voor bundel Controller + Enclosure (MSA1510i + MSA30 of MSA1510i+MSA20)

Appendix D: Hardware Overzicht

Algemeen

Dankzij vele investeringen van de Server Industrie is het lab uitgegroeid tot een waar test-lab met maar liefst 2 full-sized racks.

Bedrijven die apparatuur opsturen zijn onder andere: Intel, AMD, Sun Microsystems, HP, Supermicro, MSI & IBM. Vele apparaten werden gebruikt voor andere doeleinden dan deze masterproef. Hier volgt dus slechts een kleine bloemlezing van de apparatuur die gebruikt werd voor de benchmarks voor zowel Virtualisatie als Shared Storage.

Shared Storage

Intel SSR212MC2 (2U)

CPUs: 2 x Intel Quad-Core Xeon E5335 @ 2GHz

Geheugen: 2 x 1GB ECC-Registered PC2-5300 DDR2 (FBDIMM)

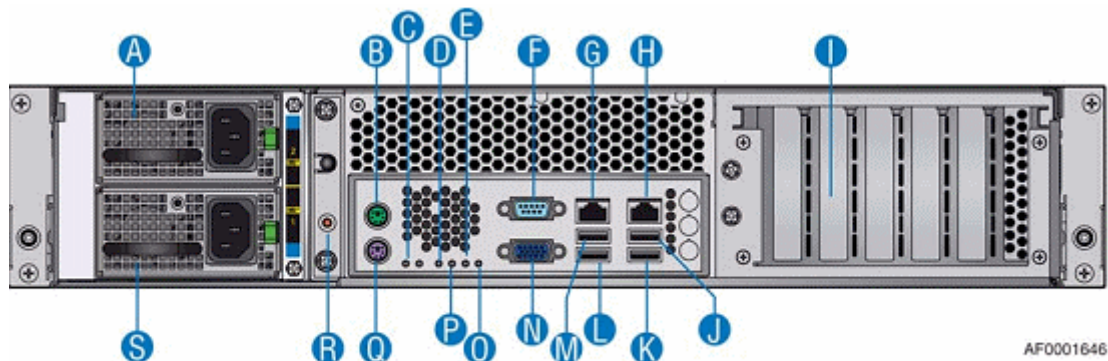
Moederbord-chipset: Intel S5000PSL motherboard

Opslag: Intern 1 x SAS 2,5" 73GB (systeem schijf) +

12 schijven Seagate Cheetah 300GB 15K5 RPM

Netwerkaart(en): onboard 1Gb/s Intel NIC, Firmware 1.03.00-0211 (Ver. 2.11)

Insteekkaart(en): SRCASAS144E met 128MB Cache



A. Power Supply Unit 2	K. USB Port 1
B. PS2 Mouse Port	L. USB Port 3
C. System Status LED	M. USB Port 2
D. MSB (POST LED)	N. Video Port
E. Bit 1 (POST LED)	O. LSB (POST LED)
F. Serial Port	P. Bit 2 (POST LED)
G. NIC Port 1 (1 Gb)	Q. PS2 Keyboard
H. NIC Port 2 (1 Gb)	R. ID LED
I. PCI Cards	S. Power Supply Unit 1
J. USB Port 0	

Figuur 116: Hardware overzicht: de Intel SSR212MC2

Gebruikte client (Tower)

CPU: Intel Pentium D 3.2GHz (840 Extreme Edition)

Geheugen: 2 x 1GB DDR2 PC2-4300

Moederbord-chipset: Intel Desktop Board D955XBK

Opslag: 80GB SATA Seagate

Netwerkkkaart(en): onboard Intel Pro/1000 PM (driver version: 9.6.31.0)

Insteekkaart(en): FC HBA Emulex LightPulse LPe1150-F4 (SCSIport Miniport Driver version: 5.5.31.0)

Fibre Channel Array (2U)

Promise VTrak E310f, FC 4Gb/s

Onboard CPU: IOP341 1.2GHz, 512MB Cache

Opslag: 12 schijven Seagate Cheetah 300GB 15K5 RPM



Figuur 117: Hardware overzicht: VTrak E310f bovenaan, Intel SRR212MC2 onderaan

Virtualisatie

Storage Rack

Promise VTrak J300S (2U)

Dual Controller, 12 SAS/SATA Disks

Opslag: 8 x Seagate Cheetah 300GB 15K5 RPM



Figuur 118: Hardware overzicht: Promise VTrak J300S

Eerste virtuele host-server: Clovertown (2U)

CPUs: 2 x Quad-Core Xeon E5345 @ 2,33GHz

Geheugen: 8 x 1GB ECC-Registered DDR2 PC2-5300 (Samsung)

Moederbord-chipset: Supermicro X7DB8, Intel 5000P (Blackford) Chipset

Opslag: SATA Seagate Barracuda 7100.10 400GB (ST3400620NS)

Netwerkkkaart(en): 2 x nVidia nForce Gigabit (82563EB)

Insteekkaart(en): LSI Logic MegaRAID SAS-controller 8344ELP,
met IOP333 CPU, 128MB Cache

Tweede virtuele host-server: Harpertown (2U):

CPUs: 2 x Quad-Core Xeon E5472 @ 3GHz

Geheugen: 4 x 2GB ECC-Registered DDR2 PC2-5300 CL5 (Crucial)

Moederbord-chipset: Supermicro X7DWN+, Intel 5400 (Seaburg) Chipset

Opslag: SATA Seagate Barracude 7100.10 400GB (ST3400620NS)

Netwerkkkaart(en): 2 x Intel Gigabit (82575EB)

Insteekkaart(en): LSI Logic MegaRAID SAS-controller 8344ELP,
met IOP333 CPU, 128MB Cache

Derde virtuele host-server: Barcelona (2U)

CPUs: 2 x Quad-Core AMD Opteron 2350 @ 2GHz

Geheugen: 4 x 2GB ECC-Registered DDR2 PC2-5300

Moederbord-chipset: Supermicro H8DMU+, nVidia MCP55 Pro Chipset

Opslag: SATA Seagate Barracuda 7100.10 400GB (ST3400620NS)

Netwerkkkaart(en): 2 x nVidia MCP55Pro Chipset gigabit controllers

Insteekkaart(en): LSI Logic MegaRAID SAS-controller 8344ELP,
met IOP333 CPU, 128MB Cache



Figuur 119: Hardware overzicht: Barcelona, Harpertown & Clovertown in hetzelfde chassis

Appendix E: ESX Troubleshooting

Algemeen

Hier een klein overzicht van de problemen die we tegenkwamen met de VMWare ESX Software en tevens hoe we ze opgelost hebben.

Enabling VMotion on different CPU's

VMotion werkt niet als de CPUs niet exact gelijk zijn, dit is (met enige risico) te omzeilen door een bestand aan te passen op de machine waar de Virtualcenter Server geïnstalleerd staat. Dit is te vinden in *C:\Documents and Settings\Alle Users\Application Data\VMware\VMware VirtualCenter* vinden we het bestand *vpxd.cfg*. Dit openen we in kladblok of wordpad en onderaan voegen we volgende regels toe boven de `</config>`:

```
<migrate>
  <test>
    <CpuCompatible>>false</CpuCompatible>
  </test>
</migrate>
```

Hierna dienen we het bestand op te slaan en de VirtualCenter Service te herstarten. Let wel op, de kans bestaat dat de virtuele machines niet meer werken na de VMotion . Dit komt dan doordat de CPUs té veel verschillen.

RDM

Je kan onder ESX wel degelijk rechtstreeks lokale schijven aan een VM hangen, zonder gebruik te maken van het VMFS. Voor lokale schijven is hier wel een console commando nodig (ESX console dus).

Dit commando zorgt ervoor dat de *gevonden* schijf gemapt wordt binnen een folder van een willekeurige VMFS systeem (bijv. waar de systeemschijven staan van de VM). Deze techniek heet RDM (Raw Device Mapping). Zoals gekend worden gevonden schijven onder de vorm van `vmhba1:1:0:0` getoond, het mappen gebeurt als volgt:

```
# vmkfstools -r /vmfs/devices/disks/vmhba1\1\0\0 /vmfs/volumes/test/sdd.vmdk
```

`sdd.vmdk` is de naam die je kiest, het pad `/vmfs/volumes/test` is het pad waar de mapping gebeurt en `vmhba1:1:0:0` is een volledige schijf, partitie's (bv: `vmhba1:1:0:1`) kun je niet mappen.

Command-line vmfs creation

Als je via de GUI een vmfs aanmaakt en je merkt dat van, bijvoorbeeld, een schijf van 2TB er slechts 200GB beschikbaar is, dan kan het helpen om de vmfs aan te maken via de command line.

Ga hiervoor naar de ESX-console (putty of rechtstreeks) en doe dan een fdisk om een vmfs-partitie aan te maken als volgt:

```
fdisk /dev/sdc
```

"n" (nieuwe partitie), "p" (primary), "1" (partitie #1), 2xEnter (defaults), "x" (expert mode), "b" (startblokken specificeren), "1" (partitie #1), "64" (aligneren), "r" (normale mode), "t" (partitie type), "1" (partitie #1), "fb" (= vmfs volume), "w" (opslaan en fdisk afsluiten)

Nu zien we dat in de folder /vmfs/devices/disks/vmhbaxxx er een vmhbax:x:x:1 bijgekomen is, dit is de nieuwe partitie die we nog moeten formateren om in te kunnen zetten. Dit kan met de vmkfstools utility:

```
# vmkfstools -C vmfs3 -b 1m -S mijnLUN /vmfs/devices/disks/vmhba1:1:0:1
```

-C is verplicht en is het type filesystem, -b is optioneel en is de blok grootte (1m, 2m, 4m & 8m mogelijk), -S is het optionele label.

KERNEL MOUNTING FAILED

Na een installatie van ESX3.5 op enkele machines met de onboard nVidia MCP55 Sata controller kregen we bij de eerste herstart een zware fout:

```
Invalid compressed format (err=2) <6>Freeing initrd memory: 7508K freed
VFS Cannot open root device "UUID=a1f28d49-8a1a-4ccf-9abf-8a38ae768a98" or 0
Please append a correct "root=" boot option
Kernel panic: VFS Unable to mount root fs on 00:00
Kernel 0xc0100000 -s .data 0xc038c000 -s .bss 0xc0443380
```

ESX3.5 maakt een xml bestand aan met een lijst van alle SATA controllers en hun id's, waarnaar hij kan verbinden om schijven te mounten. Door een bug in ESX3.5 maakt hij maar 1 controller-poort aan voor de MCP55 Sata controller waardoor we handmatig het xml-bestand moeten aanpassen. We starten hiervoor op in troubleshoot mode (service console only). En, nadat we eventuele nieuwe hardware die hij daardoor vindt geconfigureerd hebben, komen we in een console terecht:

Eerst de rechten aanpassen om te kunnen schrijven in het xml bestand

```
#chmod +w /etc/vmware/pciid/sata_nv.xml
#vi /etc/vmware/pciid/sata_nv.xml
```

Onderaan moeten we dit toevoegen, let erop dat het device met id '037e' reeds bestaat en we een device met '037f' moeten toevoegen:

```
<device id="037f">
  <vmware label="scsi">
    <driver>sata_nv</driver>
  </vmware>
  <name>MCP55 SATA Controller</name>
</device>
```

We testen of deze aanpassing correct is door volgende programma te runnen

```
#esxcfg-pciid
```

Nu zien we een aantal loading, writing & replacing lijnen.

We moeten de rechten nu opnieuw juist zetten

```
#chmod -w /etc/vmware/pciid/sata_nv.xml
```

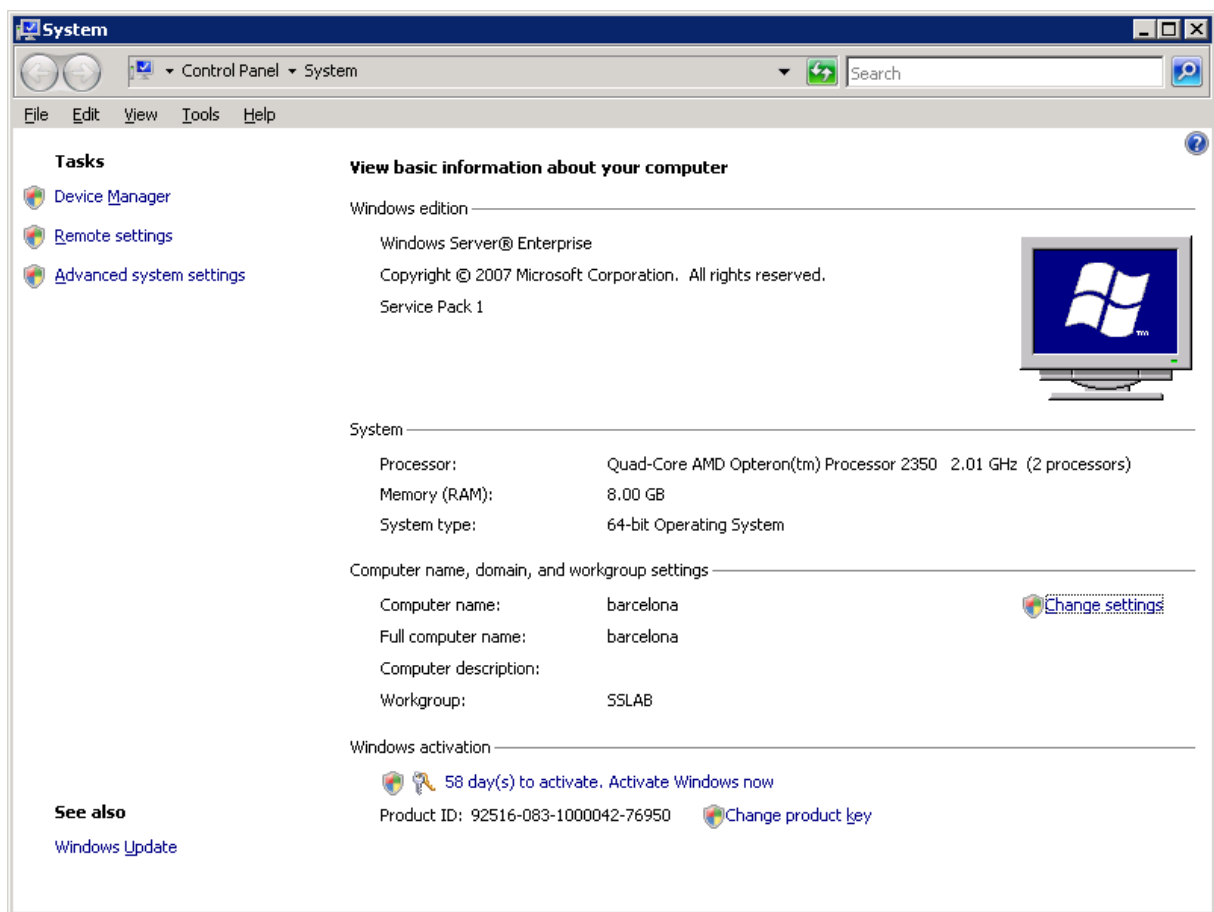
OK, nu het systeem herstarten en alles is in orde.

Appendix F: Windows Server 2008 Overzicht

Algemeen

Dit artikel werd geschreven na het bezoek aan een sprekersessie door de Microsoft Specialist Arlindo Alves op 6 November 2007 in het PIH zelf.

Windows Server 2008 is eigenlijk exact dezelfde kernel als Windows Vista, maar dan met hun nieuwste Service Pack 1 er meteen in. vandaar dat we in het Systeem Eigenschappen venster zien dat er reeds een vermelding "Service Pack 1" staat.



Figuur 120: Systeem eigenschappen van Windows Server 2008

Windows Server 2008 is meer dan een OS, het is eerder een platform, het kan verschillende zogenaamde "rollen" op zich nemen.

Standaard is er geen enkele van deze "services" geïnstalleerd, maar er zijn er 17, zoals bijvoorbeeld:

- Webservice (IIS7)
- Active Directory
- Application Server
- Terminal Services
- File, Fax, Print Server
- DHCP, DNS Server

Aan de andere kant zijn er ook features, dit zijn dan eerder applicaties of grote programma's of frameworks, er zijn er een-tal 36 beschikbaar, zoals:

- .NET Framework 3.5
- Bitlocker
- Failover Clustering
- Multipath I/O
- Message Client
- Storage Manager for SANs
- SMTP Server, SNMP Services, Telnet Client & Telnet Server
- Windows PowerShell
- Network Load Balancing

Na installatie staat Windows Server 2008 volledig dicht, d.w.z. dat de firewall aan staat, alle poorten geblokkeerd en dat Remote Desktop uitgeschakeld is. Een bedrijf moet eerst alles installeren zoals het hoort en dan pas de firewall configureren en Remote Desktop aanzetten enz...

Een kleine bespreking van enkele van de features van Windows Server 2008:

Windows PowerShell

PowerShell is een Command Line Server Setup. Het is reeds bekend dat er enorm veel bereikbaar is via de zogenaamde Command Prompt die in XP & Vista zit (cmd.exe), doch PowerShell gaat een stap verder, het ondersteunt scripting, en linux commando's, alles wat visueel kan, kan ook via de PowerShell. Linux-bash wordt ook geëmuleerd, ls & rmdir-commando's worden nu dus ondersteund.

```

Windows PowerShell
Windows PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> ls

Directory: Microsoft.PowerShell.Core\FileSystem::C:\Users\Administrator

Mode                LastWriteTime         Length Name
----                -
d-r--              8/11/2007 17:50           Contacts
d-r--             12/11/2007 16:21           Desktop
d-r--              8/11/2007 17:50           Documents
d-r--              8/11/2007 17:50           Downloads
d-r--              8/11/2007 17:50           Favorites
d-r--              8/11/2007 17:50           Links
d-r--              8/11/2007 17:50           Music
d-r--              8/11/2007 17:50           Pictures
d-r--              8/11/2007 17:50           Saved Games
d-r--              8/11/2007 17:50           Searches
d-r--              8/11/2007 17:50           Videos

PS C:\Users\Administrator> mkdir testdir

Directory: Microsoft.PowerShell.Core\FileSystem::C:\Users\Administrator

Mode                LastWriteTime         Length Name
----                -
d-----           12/11/2007 16:34           testdir

PS C:\Users\Administrator> rmdir testdir
PS C:\Users\Administrator> _

```

Figuur 121: Windows Server 2008 Powershell

Failover Clustering

In ons labo is Failover Clustering al zeer bekend, met Windows Server 2008 wordt dit nog iets verder uitgewerkt, een demo werd mij getoond met 4 Windows Server 2008 machines die virtueel draaien (op MS Virtual PC). In een VLAN werd een MSFC opgezet (Microsoft Failover Cluster), daarvoor zijn minstens 4 machines nodig.

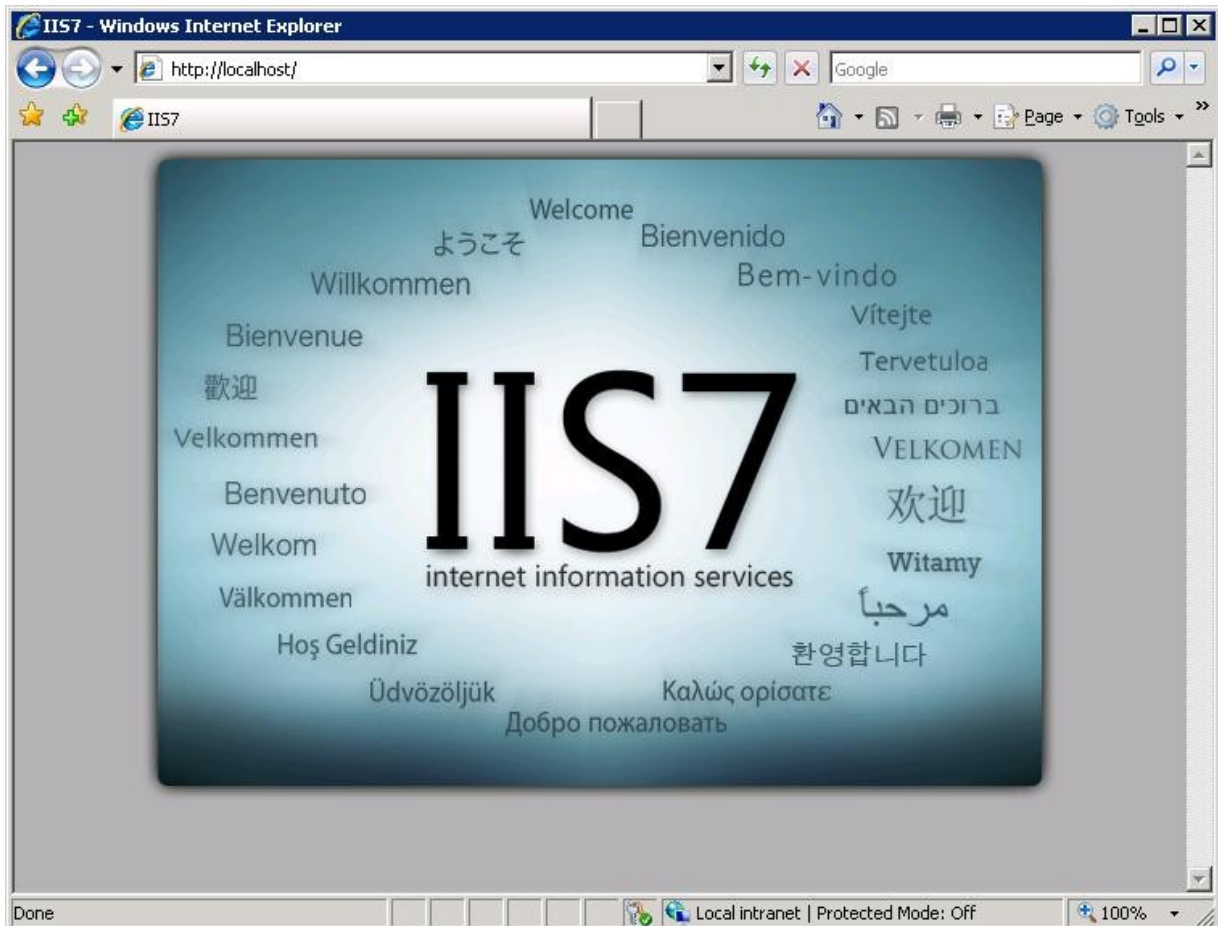
- DC: DomainController, is nodig, want MSFC kan enkel functioneren binnen een domein, deze kan uiteraard gelijk welke Windows machine zijn.
- SAN: In principe niet noodzakelijk, maar zonder SAN heeft een Cluster weinig zin, deze zorgt ervoor dat alle nodes met dezelfde data werken.
- Node1: Server met Database of Webserver op
- Node2: Server met Database of Webserver op

Alle nodes worden door middel van het heartbeat protocol gecontroleerd of ze nog online zijn. Op gelijk welken van de 2 nodes kun je een applicatie toevoegen aan de cluster, je ziet ook van de bestaande applicaties op welke node ze draaien. Je kunt ze met een muisklik verplaatsen naar de andere node, maar als 1 van de nodes plat gaat (netwerk uit, of ze worden herstart) worden de applicaties ook automatisch overgezet.

Er zijn bij het opzetten van een cluster ook validatie-tools, want het is een voordeel (vereiste?) als de 2 machines gelijk zijn qua opzet. (Patches, Drivers, Bios-upgrade ...) de validatie-tool doet deze controle.

Internet Information Services 7 (IIS7)

Deze is nu eindelijk een volwaardige rol, en kan wedijveren met Apache, hij heeft ook standaard PHP ondersteuning. Alle configuratie-gegevens worden opgeslagen in XML-bestanden en deze kunnen gekopieerd worden naar andere IIS-servers, waardoor ook deze meteen geconfigureerd zijn, er is ook een XML-bestand per website, die de website-eigenaars (op afstand) kunnen aanpassen.



Figuur 122: Windows Server 2008 Internet Information Services 7

Terminal Services

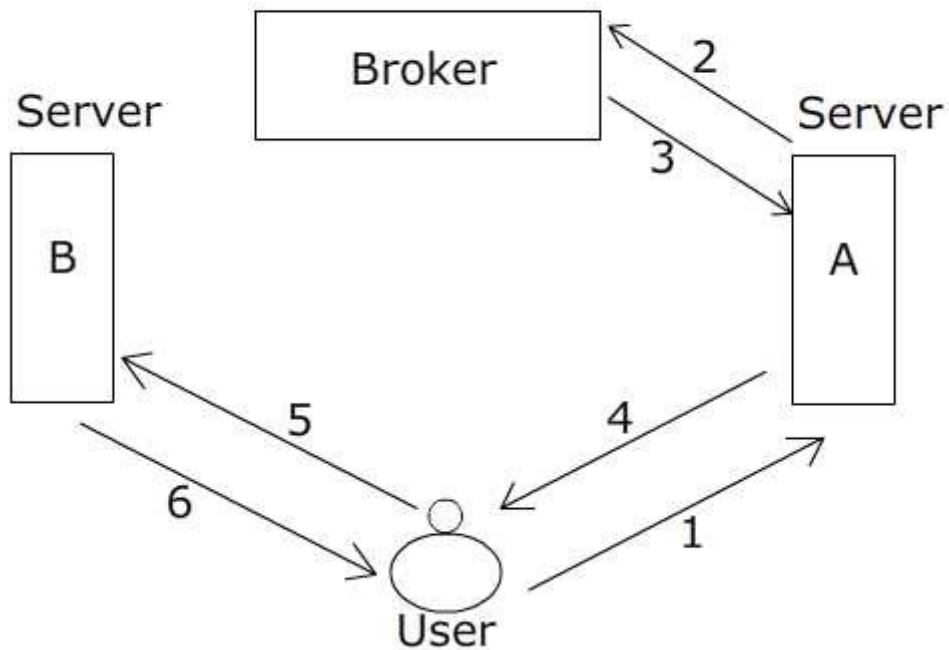
Op dit ogenblik wordt voornamelijk veel Citrix gebruikt, maar nu komt Microsoft met hun eigen Terminal Service.

Een Terminal Service zorgt ervoor dat zogenaamde "thin-clients" verbinding maken en op afstand volledige applicaties kunnen draaien, bijvoorbeeld Word 2007.

Er zijn bepaalde benodigheden, zoals dat de client's minstens Remote Desktop Client Protocol 6.0 dienen te hebben.

De Terminal Server kan via web toegang geconfigureerd worden en er is sprake van User Load Balancing, de users worden dankzij een zogenaamde "Broker" geselecteerd voor een bepaalde server aan de hand van de user Load, dit gaat als volgt:

Een user doet in stap 1 een request aan de dichtstbijzijnde server A, die vraagt zijn beurt in stap2 aan de broker met welke server de minste gebruikers verbonden zijn. Dat blijkt server B te zijn, zegt de broker in stap3, in stap4 zegt server A tegen de user dat hij moet verbinden met server B, en in stap 5 + 6 gebeurt dat ook.



Figuur 123: Terminal Services in Windows 2008: flowchart

Appendix G: Projectfiche



Projectfiche

Projecttitel:

Hardware Assisted Virtualisation

Projecttype:

Studie door Casestudy's van lokale KMO's
Uitvoering

Projectorganisatie:

Naam: Hogeschool West-Vlaanderen dep. PIH
Adres: Graaf Karel de Goedelaan 5
8500 Kortrijk
Website: pih@howest.be
www.pih.be

Projectteam:

Projectleider: Tijn Deneut
Promotor: Johan De Gelas
Copromotor: Johan Beke
Dieter Vandroemme, Karen Blancke, Geert Hofman ...

Doelstellingen:

Storage-onderzoek:

- DAS, SAN & NAS
- iSCSI vs Fibre Channel
- Shared Storage in netwerk-omgeving
- Proof of concept
- Benchmarken & Uitmeten

Virtualisatie-onderzoek:

- Zowel theoretisch als uitwerking
- Full Virtualisation (Hardware virtualisation)
- Paravirtualisation (Xen) & ESX
- Benchmarken & Uitmeten

Casestudy's:

- MCS VMWare uitbreiding
- Savaco High Availability solutions (oa Bare Metal Storage)

Kwaliteitseisen:

- Wettelijke voorzieningen: n.v.t.
- Veiligheidsnormen: n.v.t. (zie risico-analyse)
- Milieueisen: n.v.t.
- Toepasbaar binnen KMO's
- Verder wel rekening houden met TCO (Total Cost of Ownership) die via tabellen kan worden opgemaakt.

Input:

- Kennis, ervaring & raadgeving van de externe en vooral interne promotor
- Uitgebreide database van eBooks, Wikipedia, Google ...
- KMO's die paraat staan met concrete consult, internationale bekendheid

Output:

- Realiseren van de geformuleerde doelstellingen
- KMO consult -> uitwerking Casestudy
- Virtualisatiemodellen die toepasbaar zijn binnen KMO's
- Medewerking aan internationale publicatie's met feedback tot gevolg
- Kennisdifusie naar Vlaamse KMO's via IT Pro
- Scriptie en thesisverdediging

Projectbeperkingen:

- Geen Security of Backup analyses
- Geen diepgaande studie over processen die niet rechtstreeks met het eindwerk te maken hebben

Projectmijlpalen:

Nr.	Mijlpaal	Periode
1.	Lezen eindwerken voorgangers, documentatie	Begin September
2.	Kennismaking & Benchmarking Storage, netwerkanalyse	September – Oktober

3.	Casestudy MCS, Virtualisatie onderzoek	November, December
4.	Casestudy Savaco, Benchmarking Virtualisatie	December-Januari
5.	Theoretische studie's Para- & Full virtualisatie	Februari-Maart
6.	Toepassen Virtualisatie op Database (Clustering) & Webserver (Network Load Balancing)	Maart-April
7.	Schrijven inhoudsopgave	April
8.	Thesis schrijven: 1 ^e versie	April-Mei
9.	Maken samenvatting & Opstellen eindversie	Mei-Juni
10.	Vorbereiden thesisverdediging	Juni

