Vierdelige serie begint met overzicht van virtualisatie

Databases in virtuele datacenter-omgevingen (1)

Bram Dons

Er woedt een ware virtualisatiegolf in IT-land. Veel leveranciers schreeuwen deze 'nieuwe' technologie van de daken en proberen ons te overtuigen van het nut. Virtualisatie van server-systemen, ook wel 'consolidatie' genoemd, heeft zoals elke andere technologie echter voor- en nadelen.

De toepassing van virtualisatie moet dan ook geen doel op zich zijn, maar alleen worden toegepast wanneer dat werkelijk zinvol is en de voordelen groter zijn dan de nadelen. Naast e-mail servers zijn databases de meest kritische applicaties in bijna elke onderneming. Bij de toepassing van beide applicaties in een virtuele omgeving is het extra oppassen bij de keuze voor een bepaalde virtualisatietechnologie. De keus is vandaag de dag groot omdat veel databases tegenwoordig op een gevirtualiseerd systeem kunnen draaien. Een ander aspect betreft het verschil in ontwikkelingsstadium van de diverse virtualisatietechnologieën. Daarnaast worden gebruikers overspoeld met allerlei begrippen en definities over virtualisatie die door de leveranciers van virtualisatie worden gehanteerd. IT-beslissers staan voor de moeilijke taak om een keuze te maken uit de verschillende virtualisatieproducten en -technologieën. Voor het maken van de juiste keuze is een goed inzicht vereist in de daaraan ten grond liggende technologie. Het is daarom dat we in dit inleidend artikel over de toepassing van databases in een virtuele omgeving trachten een overzicht te geven van de meest toegepaste, op x86/x64 architectuur gebaseerde, virtualisatietechnologieën.

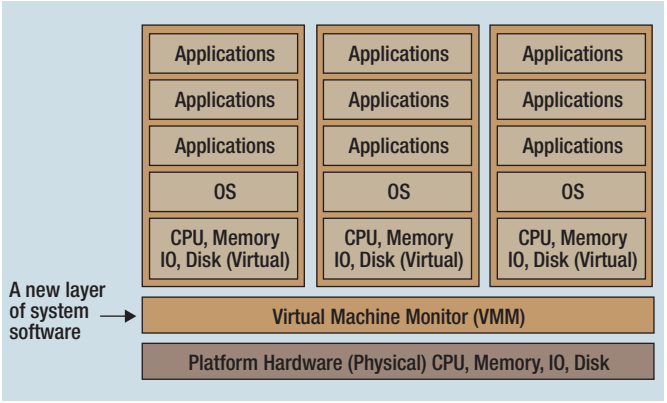
Voor- en nadelen virtualisatie

Virtualisatie biedt een aantal voordelen. Iedere IT-beheerder kan ze tegenwoordig wel uit het hoofd opsommen. Toch noemen we nog even de belangrijkste: betere bezettingsgraad van fysieke servers door server-consolidatie (waardoor minder hardware nodig is en dus een lagere TCO), real-time migratie van processen, eenvoudiger beheer en goede backup- en recovery-mogelijkheden. Een voordeel waar leveranciers mee schermen is een vereenvoudigd beheer, maar de praktijk heeft inmiddels wel geleerd dat de complexiteit van een virtuele omgeving exponentioneel toeneemt met het aantal VM's, en wordt server-virtualisatie nog eens gecombineerd met storage-virtualisatie dan neemt de complexiteit nog eens met een factor x toe. Een ander aspect waaraan vaak voorbij wordt gegaan, is de beveiliging van een gevirtualiseerde omgeving. Zoals is gebleken brengt virtualisatie een aantal onvoorziene beveiligingsrisico's met zich mee.

Historisch overzicht van virtualisatie

Gebruikers worden overspoeld door allerlei termen, begrippen, definities rondom virtualisatietechnologie: applicatievirtualisatie, hypervisor, para-virtualization, full virtualization, emulation, instruction set virtualization, binary translation, Hyper-V, host-based, OS virtualization en hardware-assisted virtualization. Welke gebruiker weet nu precies het onderscheid daartussen voor, vaak dezelfde, virtualisatietechnieken en de betekenis daarvan? Daarom terug naar de basis, en zien we wat virtualisatie in zijn oorspronkelijke vorm eigenlijk was en hoe de techniek de laatste jaren is uitgegroeid tot wat het nu is.
Virtualisatie is geen nieuwe technologie, want het stamt al uit 1960 toen IBM met de System 360 Model 67 mainframe alle hardware-interfaces virtualiseerde via een Virtual Machine Monitor (VMM), zie afbeelding 1.
In de begindagen van het computertijdperk heette het OS een *supervisor*. De mogelijkheid om een Operating System op andere

In het eerste deel van een vierdelige serie over virtualisatie geeft Bram Dons een overzicht van de belangrijkste virtualisatietechnologieën voor toepassing van databases in de datacenter. De daarop volgende drie artikelen behandelen elk de afzonderlijke virtualisatie-implementaties in de relatie tot databases; in deel twee komt de Xen-technologie aan de orde, in deel drie VMware en tenslotte in deel vier Hyper-V.



Afbeelding 1: Virtual Machine concept.

OS's te draaien, introduceerde al in jaren zeventig de term *hypervisor*. Na de VMM-technologie volgde de emulatie-technologie waarin het gedrag van een complete computer werd gekopieerd naar een software-programma. De emulatielaag 'praatte' tegen een OS dat op zijn beurt weer met de computer hardware praatte; twee populaire open source emulators waren QEMU en Bochs.

Een van de meest belangrijke eigenschappen van emulatie is dat alle hardware wordt geëmuleerd, ook de CPU. Dit heeft als voordeel dat een voor een bepaalde architectuur ontwikkeld OS ook op een andere kon draaien. Nadeel was dat de CPU extra werk moest verrichten, wat zich vertaalde in een lagere prestatie. Bij de volgende generatie was er geen host OS meer nodig tussen VM's en de hardware, maar werd een VMM geïmplementeerd. Die kennen we tegenwoordig als 'hypervisor' en maakte virtualisatie een stuk efficiënter.

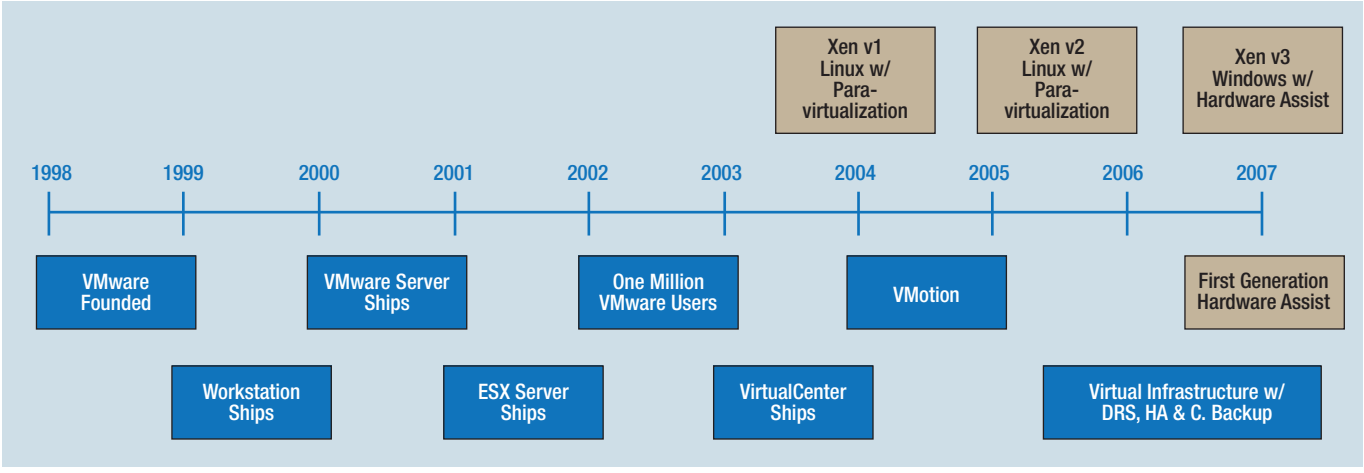
De noodzaak van virtualisatie op OS-niveau is vandaag de dag ontstaan als resultaat van een vreemde toevalligheid van krachten die in de markt speelden. Ten eerste, software-applicatie-architecturen worden steeds complexer en zijn gebaseerd op multi-threaded, multi-process en multi-tiered systemen. Deze zijn moeilijk schaalbaar, te configureren en beheren. Ten tweede, de adoptie van de zogenaamde 'scale-out' computing infrastructure, gebaseerd op low-cost industriestandaard servers heeft geleid tot een grootschalige implementatie van server hardware.

De moderne hypervisor

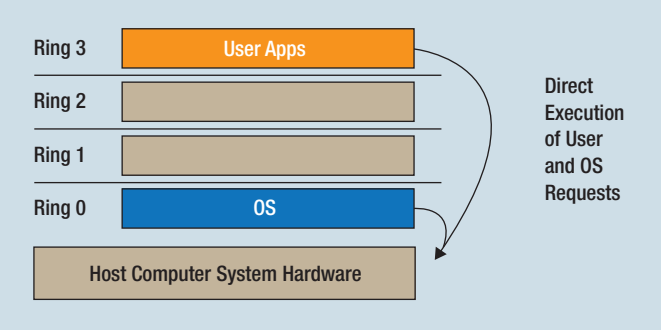
De komst van de hypervisor heeft alles veranderd, want de noodzaak voor een host OS kwam daarmee te vervallen. Althans, er is geen host OS meer in een vorm die de meesten van ons zouden herkennen. Om niet het wiel uit te moeten vinden, leenden de meeste hypervisor-implementaties wel iets van Linux, Unix of Windows. De hypervisor-methode biedt een aantal potentiële voordelen, met in de eerste plaats betere prestaties. Hoewel een hypervisor zelf niet veel bijdraagt aan de beveiliging op individueel VM-niveau, biedt de onderliggende hypervisor-code voor hackers minder aanknopingspunten (vanwege de kleine 'footprint'), in vergelijking met een standaard host-gebaseerd OS. De toepassing van een hypervisor kent ook nadelen, zo is deze meer hardware-specifiek.

Hypervisor-benaderingen

De x86-gebaseerde OS's zijn ontworpen om direct op de 'bare-metal' hardware te draaien en er vanuit te gaan dat ze de volledige 'eigenaar' zijn van de computer hardware. De x86-architectuur biedt vier *privileged* niveaus, beter bekend als Ring 0, 1, 2 en 3. Applicaties draaien in Ring 3 en het OS heeft alleen via 'privileged' instructies toegang tot het geheugen en de hardware in Ring 0, zie afbeelding 3. Voor virtualisatie van de x86-architectuur is er een extra virtualisatielaag nodig onder de OS Ring 0 laag, de zogenaamde 'Ring -1' en dient voor creatie en beheer van VM's en ondersteuning van 'shared resources'. Sommige 'sensitive' instructies kunnen niet effectief worden gevirtualiseerd, omdat ze een verschillende semantiek hebben wanneer ze niet in Ring 0 worden uitgevoerd. De moeilijkheid om deze sensitive en privileged instructies af te vangen en te vertalen vormde voor de ontwerpers een hele uitdaging en voor lange tijd leek virtualisatie op de x86-architectuur daardoor een onmogelijkheid. VMware slaagde er echter in 1998 toch als eerste in om de x86 CPU te virtualiseren.



Afbeelding 2: Ontwikkeling x86 virtualisatietechnologieën (bron VMware).



Afbeelding 3: Binary translation x86 (bron VMware).

OS-virtualisatie wordt mogelijk gemaakt door een laag met systeem-software aan te brengen, de zogenaamde hypervisor, tussen een zogenaamd 'guest' OS en de onderliggende fysieke hardware. Deze software-laag is er verantwoordelijk voor dat meerdere VM's de hardware-bronnen op een enkele server kunnen delen. Elk guest OS denkt dat het de exclusieve gebruiker is van alle fysieke bronnen, maar in werkelijkheid moet het deze delen met de andere gasten. De hypervisor is er verantwoordelijk voor om transparant alle fysieke bronnen (memory management units, I/O-devices, DMA controllers, enzovoort) zo eerlijk mogelijk over alle guests OS's te verdelen en de guests een gevirtualiseerde abstractie van de fysieke bronnen te bieden.
De x86-architectuur is de meest toegepaste computerarchitectuur in het enterprise datacenter. Maar deze architectuur was oorspronkelijk niet ontworpen voor virtualisatie, met als gevolg dat een hoogpresterende virtualisatieoplossing moeilijk te bereiken was. De nu twee toonaangevende software-virtualisatiemethoden voor het datacenter zijn 'full virtualization' en 'para-virtualization'. Daarnaast zijn er de AMD-V en Intel VT processoren, zogenaamde 'hardware assisted', virtualisatietechnologieën. Beide leveranciers hebben een aantal aanpassingen aan de x86-architectuur en instructieset in de CPU aangebracht om de prestaties bij virtualisatie te verbeteren.

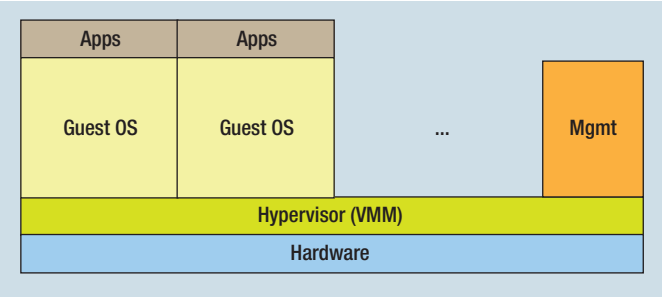
Full virtualization

Full virtualization vertegenwoordigt de eerste generatie x86/x64 server-virtualisatie en maakt van de meest directe methode gebruik om het onderliggende hardware-platform in software na te bootsen. Van alle instructies vormen de exclusief door het OS gebruikte instructies (de zogenaamde 'privileged' instructions voor interrupt-afhandeling, lezen en schrijven naar devices en virtueel geheugen) een belangrijke groep instructies die geëmuleerd moet worden. Per definitie kunnen deze instructies niet door een gebruikersapplicatie worden uitgevoerd. Een toegepaste techniek is om de emulatie-software te dwingen deze als gebruikerscode in een VM, inclusief de gevirtualiseerde OS, uit te voeren.

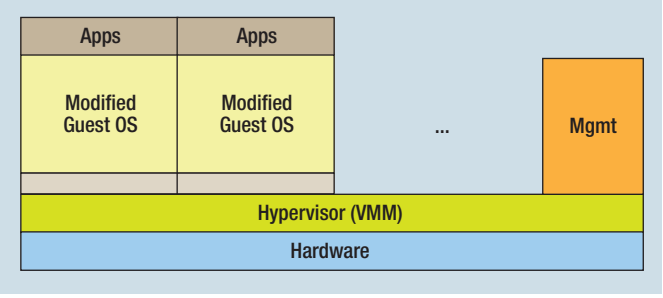
De x86-architectuur is vanuit een vroeger ontwerp geëvalueerd, waarbij geen rekening is gehouden met het gebruik van virtualisatie. Bij toepassing van full virtualization resulteert dit vaak in zogenaamde 'virtualization holes'. Bepaalde in CPU 'user' of 'supervisor' mode uitgevoerde instructies kunnen verschillende resultaten opleveren. Een techniek om dat te voorkomen is het scannen van OS-code en deze instructies vervangen. Deze techniek, 'binary translation' wordt uitgebreid toegepast in VMware Server en maakt in principe een direct gebruik van elke type guest OS mogelijk, zie weer afbeelding 3. Het scannen van code, voorafgaande aan de uitvoering daarvan, brengt nogal wat CPU-overhead met zich mee en vermindert de applicatieprestaties. Voordeel van deze 'full virtualization' methode is dat een guest OS en de bijbehorende applicaties ongemodificeerd in een virtuele omgeving kunnen worden gedraaid en het is de enige virtualisatiemethode waarvoor het OS niet hoeft te worden aangepast. Het guest OS is er zich dan ook totaal niet van bewust dat het in een virtuele omgeving draait. Naast de VMware-producten is Microsoft Virtual Server het meest bekende en toegepaste virtualisatieproduct dat is gebaseerd op een 'full virtualization' architectuur. Daarbij draait de virtualisatie software-laag bovenop een bestaand OS, bij VMware is dat Linux en bij Virtual Server is dat Windows. Andere producten in deze virtualisatiecategorie zijn onder meer: VMware Workstation, Parallels, Linux KVM en de Virtualbox van SUN.

Para-virtualization

Een alternatief voor de full virtualization is om de machine-instructies niet te vertalen, maar een aanpassing van het OS waarbij de ingepaste instructies zijn geoptimaliseerd voor gebruik in een virtuele omgeving. De techniek heet 'para-virtualization' of 'OS Assisted Virtualization'. Het guest OS wordt naar een geïdealiseerde hardware-laag overgezet en alle hard-

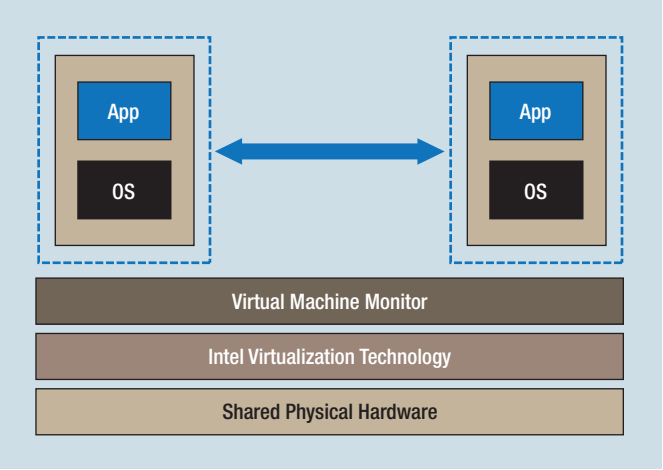


Afbeelding 4: Full Virtualization architectuur (bron IBM).



Afbeelding 5: Para-virtualization architectuur (bron IBM).

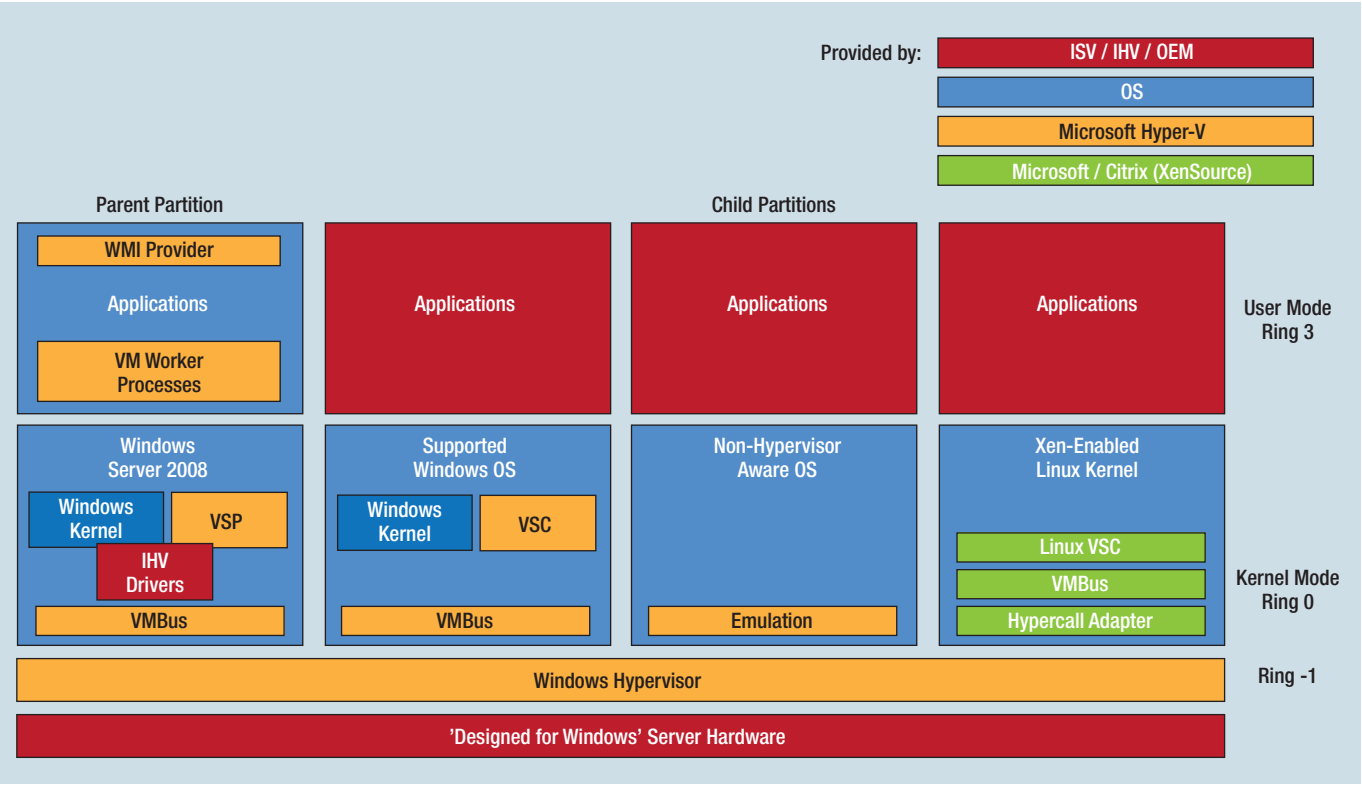
ware-interfaces worden gevirtualiseerd. Bij wijziging van het OS aan bepaalde hardware-datastructuren (zoals page tables) roept het een API in de hypervisor aan. De hypervisor houdt alle aangebrachte wijzigingen bij en beslist op welke optimale manier bij elke context switch de hardware moet worden aangepast.
Bij deze meest efficiënte methode voor virtualisatie maakt het OS gebruik van een speciale API om met de hypervisor te communiceren, de laatste is uiteindelijk verantwoordelijk voor de afhandeling van de virtualization requests voordat ze naar de hardware worden doorgestuurd. Door deze speciale API hoeft de hypervisor geen extra tijd meer te besteden aan het vertalen van instructies, zoals dat bij full virtualization het geval is. In principe zijn met para-virtualization bijna dezelfde prestaties te bereiken als met een niet-gevirtualiseerd systeem, maar de behaalde prestatievoordelen zijn wel afhankelijk van de werkbelasting.



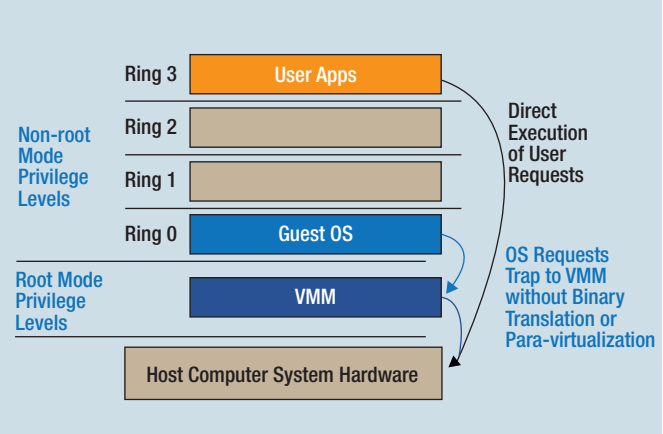
Afbeelding 7: Implementatie hardware-assisted virtualisatie (bron Intel).

VM guest OS's worden 'para-gevirtualiseerd' door de OS-kernel te hercompileren en de installatie van kernel mode drivers. Naast dit belangrijk nadeel van para-virtualization vraagt het ook van de software-leverancier om een aanzienlijke en continue ondersteuning van de guest OS's. Voor open source gebaseerde OS's, zoals Free BSD, Linux en Solaris, is dit geen probleem maar voor gesloten OS's moet de para-gevirtualiseerde hypervisor vertrouwen op de hardware-ondersteuning door de leverancier. Voor sommige OS's is het niet altijd mogelijk om een volledige para-virtualization door te kunnen voeren, sommige ongemodificeerde OS's (bijvoorbeeld Windows 2000/XP) bieden een slechte compatibiliteit en een slechte portabiliteit. Para-virtualization elimineert de noodzaak voor binary translation en is dan de binaire vertaalmethode. In combinatie met 'hardware-assisted' technologie biedt het zo goed als 'near-native' prestaties.

De para-virtualization markt wordt aangevoerd door de Xen-gebaseerde producten. Microsoft is nieuwkomer op de para-virtualization markt en heeft om prestatieredenen de Xen-architectuur als voorbeeld gebruikt voor zijn nieuwe Windows Hypervisor, 'Hyper-V' genaamd, op de Windows Server 2008. VMware wordt vaak in de media onterecht onder de para-gevirtualiseerde oplossingen geschaard. De verwarring komt omdat VMware al jaren bepaalde aspecten van para-



Afbeelding 6: Microsoft Hyper-V architectuur (bron Microsoft).



Afbeelding 8: Hardware-assisted benadering x86 (bron VMware).

virtualization technieken gebruikt in haar producten, in de vorm van VMware tools en geoptimaliseerde virtuele device drivers. De VMware tools service en vmxnet device driver zijn echter geen CPU-gebaseerde para-virtualization oplossingen! Het zijn minimale in het guest OS 'non-intrusive' geïnstalleerde aanpassingen, waarvoor de OS-kernel niet hoeft te worden aangepast.

Hyper-V architectuur

Microsoft's Hyper-V verschilt nogal van de andere bestaande hypervisors. Want de meeste hypervisors zijn gebaseerd op Linux- of Unix-code, terwijl Hyper-V nauw verbonden is met het Windows Server 2008 OS. Maar er zijn nog meer verschillen. Bijvoorbeeld, hoewel Hyper-V direct op de server hardware wordt geïnstalleerd, waardoor het meerdere VM's kan huisvesten in logische aparte 'partitions', bestaat de primaire partitie (ook wel 'parent' partition genoemd) uit een Windows Server 2008 kernel. Hyper-V kan daardoor handig gebruik maken van de bestaande drivers en services binnen de bestaande Windows software, waardoor het beter met de hardware kan communiceren en een goede beheerondersteuning kan bieden. Microsoft zelf beschrijft Hyper-V als een 'microkernelized' architectuur met een minimale 'footprint' (slechts 50.000 coderegels). De Hypervisor draait in de zogenaamde 'Ring -1' mode, de diverse OS's in Ring 0 en de Applicaties in Ring 3. De geboden Hyper-V functionaliteit komt nog maar voor een deel overeen met Microsoft's grootste concurrent, de VMware ESX Server. Want een belangrijke feature, de real-time overzetting van VM's onder VMware ESX, 'VMotion' genaamd, ontbrak tot voor kort in Hyper-V. Microsoft heeft wel onlangs de bèta-versie van Hyper-V versie 2 voor evaluatie vrijgegeven, waarin een soortgelijke functionaliteit, 'Live Migration' genaamd, wordt ondersteund. De verwachting is dat pas op zijn vroegst eind 2009 de officiële release van Windows Server 2008 R2 plaats zal vinden, waarin opgenomen Hyper-V 2 met onder meer daarin Live Migration.

Hardware-assisted virtualization

CPU hardware-leveranciers zijn bezig met de ondersteuning van x86-virtualisatiearchitectuur. De eerste generatie verbeteringen betreft Intel's Virtualization Technology (VT-x) en AMD's AMD-V, met als doel de privileged instructies van een nieuwe uitvoerende CPU mode te voorzien, zodat de VM in een nieuwe 'root mode' onder Ring 0 kan draaien. De privileged en sensitive instructie-aanroepen worden automatisch naar de hypervisor gedirigeerd en vervalt de noodzaak voor binary translation of para-virtualization. Hyper-V en Xen maken hier al wel gebruik van, VMware maar slechts in beperkte mate: 64-bit ondersteuning van guests op Intel-processoren. Naast virtualisatie van de CPU zijn het geheugen, devices en I/O de volgende te virtualiseren hardware-componenten. Intel heeft daartoe een viertal chips uitgebracht, VT-x (Xeon) en VT-i (Itanium) voor processoren, VT-d voor Directed I/O en VT-c voor Connectivity.

Databases en virtualisatie

Hiervoor gaven we een eerste indruk van de twee belangrijkste stromingen in virtualisatieland; full- en para-virtualization. Bij de uiteindelijke keus voor een gevirtualiseerd platform en database spelen nog diverse andere zaken een rol.
Ten eerste de vraag op welk OS en welk virtueel platform, welke database wordt ondersteund. We zagen dat, afhankelijk van het type virtualisatie, er aanpassingen nodig zijn voor elk te implementeren OS. Want lang niet alle OS's worden op alle virtuele platforms ondersteund, wat weer de mogelijkheid beperkt om een bepaald type database te implementeren.
Een andere overweging heeft te maken met de lange-termijn-strategie. De diverse virtualisatieproducten bevinden zich nog in verschillende ontwikkelingsstadia. Zo is VMware het verst gevorderd met zijn ESX Server en heeft Microsoft nog een behoorlijke achterstand met Hyper-V. Wil men nu investeren in VMware's full virtualization-oplossing, of wachten op Microsoft's Hyper-V, of de al bestaande Xen para-virtualization implementaties van bijvoorbeeld Citrix of Oracle? Kiest men voor de open Xen-omgeving, met zijn beperkte ondersteuning van OS's, of voor de gesloten VMware-omgeving, maar wel weer met een brede ondersteuning van OS's?

Er schuilen diverse gevaren, niet alleen van vendor lock-in, in de keuze voor een bepaalde virtualisatietechnologie. Welke technologie zal standaard worden: full- of para-virtualized, of zullen beide naast elkaar voortbestaan? Wat zijn de mogelijkheden als men nu kiest voor een bepaalde technologie en men wil later toch overstappen op een andere? Maakt het virtuele disk format waarin VM's zijn opgeslagen deze overstap daarna nog mogelijk? Het zijn allemaal factoren die moeten worden meegenomen bij het nemen van een beslissing bij de invoering van databases in een virtuele omgeving. In drie vervolgartikelen gaan we nader in op voorbeelden van de twee genoemde virtualisatietechnologieën, full en para-virtualization, en zien we praktische implementaties van databases in elk van deze gevirtualiseerde omgevingen.

Bram Dons is onafhankelijk IT consultant.