Xen, van klein tot groot (deel 1)

hands on ervaringen met kleine systemen

(NLUUG 10 mei 2007, Luc Nieland)

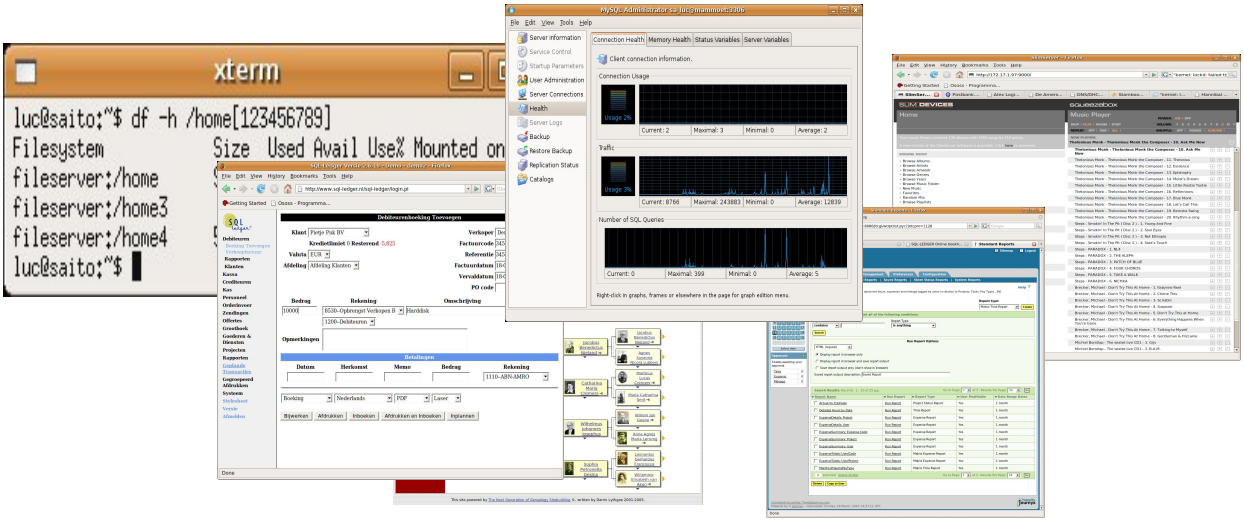
Sinds enkele jaren verrijkt Xen, een open source hyperviser product, het nog beperkte speelveld van besturingssysteem virtualisatie. De vraag is natuurlijk; is het ook bruikbaar voor productiesystemen, ook als deze serieus belast worden, en ook indien het volcontinue gebruik betreft? Dit is deel 1 van twee presentaties die vanuit het perspectief van twee verschillende doelgroepen de inzetbaarheid van Xen behandelen.

Onderwerp in dit deel zijn ervaringen met Xen vanaf release 2 in de afgelopen twee jaar met speciale test opstellingen en enkele productie-omgevingen. Alle met relatief bescheiden hardware (een tot twee x86 cores) en bijpassend gebruik van standaardapplicaties als mail-, web- en databasesystemen. Naar aanleiding van concrete setups staan we inhoudelijk stil bij vragen zoals; "hardware-sizing", distributiekeuze, kernel verkrijging, 32 of 64-bit? Aanvullend daarop wordt ook aandacht besteed aan enkele valkuilen en meer geavanceerde functionaliteit als migratie-clustering met bijbehorende "shared-storage" dilemma's.

Hoewel Xen inmiddels meerdere besturingssystemen ondersteunt, blijft de scope beperkt tot Linux. Aansluitend op deze presentatie gaat deel 2 verder met de ervaringen met grotere systemen tot wel 64 CPU cores.

Inleiding

Deze paper beschrijft de ervaringen met virtualisatie van Linux server systemen gedurende de periode 2005-2007 op relatief bescheiden hardware met 1 of 2 CPU's. Het betrof testopstellingen en opstellingen in kleine organisaties van tot 20 personen. De benodigde functionaliteit liep uiteen maar was over het algemeen een selectie uit de verzameling: firewall, fileserver, e-mail server, webserver, database server, vpn-server, boekhouding, tijdsverandwoording en streamingaudioserver. De scope van deze paper beperkt zich daarbij tot alle benodigde functionaliteit die middels software op linux werd gerealiseerd.



Hoewel Unix/Linux in principe alle mogelijkheden bied om meerdere functionaliteiten middels meerdere stukken software veilig tezamen op een machine met een operatingsysteem te realiseren, is het om uiteenlopende redenen soms gewenst om deze toch te scheiden over meerdere besturingsysteem instances. De scheiding is daarmee nog groter. Dit is een discussie apart, maar er valt wat voor te zeggen, denk alleen al aan versies en gedeelde libraries. Zonder virtualisatie betekend opdeling van applicaties dus dat er meerdere fysieke machines nodig zijn. Met de hoge capaciteit van de huidige x86-hardware, betekend dit dat er vaak veel overcapaciteit ontstaat wanneer deze opdeling van applicaties wordt toegepast. Veel overcapaciteit betekend meestal wel een prima performance voor de applicatie, maar het betekend wel meer kosten. Daarnaast zijn er altijd wel systemen, waar er nu net even weer te weinig capaciteit is. Kortom, ondanks de kosten, is het nog niet eens altijd perfect. Met virtualisatie kunnen deze twee problemen verminderd worden. De vraag is; welk operatingsysteem virtualisatie systeem hiervoor te kiezen?

Medio 2005 was een bekend virtualisatiesysteem voor linux systemen; User-mode-linux (UML). Hierbij wordt er middels een kernel patch de mogelijkheid geboden om een linux-kernel, en daarmee feitelijk een virtuele Linux machine, als proces bovenop op een gewone linux machine te draaien. Dit virtualisatiesysteem is enige tijd getest en gebruikt, en werkte goed op licht belaste systemen. Een groot nadeel van UML was en is echter het aanmerkelijke verlies van performance in de gevirtualiseerde systemen. Enigzinds begrijpelijk, want het was vooral ontworpen om de feature, en niet om de performance. Naar het schijnt werd het bijvoorbeeld door kernel ontwikkelaars behoorlijk gebruikt.
Het medio 2005 gelanceerde Xen in versie 2 vertoonde voor wat betreft gebruik in eerste instantie dan ook grote gelijkenis met UML. Ook hier werd middels kernel aanpassingen hetzelfde doel bereikt. De interesse was dus snel gewekt...

Xen bestaat kortweg uit een een stukje software die de hypervisor vormt, twee kernel patches, en bijbehorende userspace tools. De virtualisatie wordt gerealiseerd middels de standaard ring 0,1,2 en 3 modes van de X86 processor. Een concept wat in de meeste processoren zit. In de Xen situatie neemt een hypervisor de supervisor mode , ofwel de eerste plaats op de processor in (ring 0). Vervolgens wordt er in ring 1 een gepriviligeerd operatingsysteem gestart, het domein-0 (dom0). Alleen hiervandaan kan nog alle hardware benaderd worden. Van hieruit wordt ook de hypervisor bedient, en worden de zogenaamde para gevirtualiseerde "unprivileged" machines gestart (domU).
De hypervisor is een product van de Xen ontwikkelgroep, evenals de patches om een dom0 of domU kernel te maken van een standaard Linux kernel. Er zijn voor de dom0 en de domU afzonderlijke patches, maar deze zijn ook tegelijk toe te passen, zodat een kernel ontstaat die zowel als dom0 en als domU te gebruiken is. Dit is wat tegenwoordig meestal wordt gedaan wegens gebruiksgemak.

De operatingsystemen die momenteel met Xen in paravirtualisatie mode te gebruiken zijn, zijn voor de dom0 Linux en sinds kort solaris-x86. Als domU zijn Linux netBSD, FreeBSD, Solaris x86 en zelfs MS windows-NT beschreven. Belangrijk hierbij is dat de kernelpatch van MS windows-NT niet te koop is en dus puur van wetenschappelijke waarde is. Zoals al genoemd heeft een domU dus geen volledige hardware access; netwerk en disk-i/o van een domU gaan via de dom0. Volgens beschrijvingen kan de domU wel exclusief een PCI-device of een seriele port toebedeeld krijgen. Dit kan van nut zijn voor bijvoorbeeld een isdn kaart voor een software VOIP-centrale of een netwerkkaart voor een firewall.

Hoe is Xen gebruikt

Deze paper beschrijft alleen de ervaringen met de zogenaamde paravirtualisatie mode van Xen. Dit is de oorspronkelijke verschijning van Xen. Deze mode werkt op alle X86 processoren en vereist geen speciale hardware. Wel vereist deze mode voor zowel de dom0 als de domU kernels aanpassingen. De beloning is dan weer dat deze mode door het Xen team is beschreven als de best performende mode.

De basis Xen begint met een fysieke machine, waarop normaal een operatingsysteem wordt geinstalleerd. Dit operating systeem wordt na toevoeging van de Xen software de dom0.
Hiertoen werd normaal gesproken een zogenaamde “minimaal geinstalleerde” Debian GNU/Linux gebruikt. Dit wegens persoonlijke voorkeur, ervaring en goede resultaten. Met name de mogelijkheid van Debian om een behoorlijk minimaal systeem te maken, kwam in deze constructie goed van pas. Zoals aangeraden door het Xen-team, wordt op de dom0 niets meer gedaan dan configureren en managen van de domU's. Des te minder software er op de dom0 staat, des te minder hoeft er aan de dom0 onderhouden te worden was daarin het motto.

De Xen toevoegingen om tot een dom0 te komen waren ten tijde van versie 2, op twee manieren te realiseren. Eenerzijds met kan-en-klaar gecompileerde x86-32 kernels van Cambridge-labs. Anderzijds met behulp van de sources voor de hypervisor, de patches en de userspace tools. Voor installatiegemak werd een make script meegeleverd waarmee alles gecompileerd en geplaatst kon worden.
De tryout opstellingen (versie 2.x) werden snel met behulp van de binary-kernels van cambridge labs gemaakt. Voor betere hardwareondersteunde en geoptimaliseerde opstellingen werden al snel zelfgemaakte kernels gebruikt middels de Cambridge-labs scripts.

Sinds ongeveer een jaar dook in veel Linux distributies langzamerhand ook de Xen software op. Soms al standaard meegelevert, soms via de uitbreidings repositories, meestal eerst via extra repositories. Zo ontstonden er ruim een jaar geleden een in een van de extra repositories al xen-3.0.1 packages voor Debian-3.1. Deze zijn inmiddels doorontwikkeld en verhuist naar de standaard repository voor Debian-4.0. Deze Xen packages bleken dermate prima te voldoen en dermate goed op de distributie aan te sluiten dat deze het afgelopen jaar vrijwel zonder uitzondering zijn gebruikt. Het installatie en onderhouds gemak speelde in die keuze natuurlijk een niet onbelangrijke rol.

Hands-on: een kernel voor de dom0
Het maken van een Xen systeem begint met een gepriviliseerde host. Deze wordt bijvoorbeeld gemaakt door op een fysieke machine met een standaard Linux de Xen software toe te voegen. Het verschilt per distributie, een SUSE 10 server systeem doet dit toevoegen bijvoorbeeld al standaard, er hoeft slechts van kernel gewisseld te worden om een Xen dom0 in handen te hebben. Een Debian systeem is iets meer werk, maar indien er gekozen wordt om een beetje op de toekomst vooruit te lopen en versie 4.0 wordt gebruikt, dan is dit te doen middels het installeren van slechts enkele packages uit de repository. Even hands on, gaat dit met de commando's:

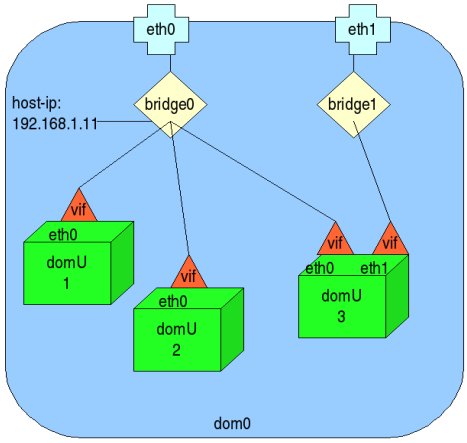
```
apt-get install iproute bridge-utils
apt-get install xen-linux-system-2.6.xx-x-xen-amd64    (2.6.xx-x is 2.6.18-4 march 2007)
```

Alle benodigde zaken worden verder middels depencies afgehandeld.

In de grub config /boot.grub/menu.list is nu een interessante toevoeging te zien:

```
title      Xen-3.0 X86-64 / XenLinux 2.6.12.6-xen0
           kernel     /boot/xen-3.0.gz  dom0_mem=131072
           module     /boot/vmlinuz-2.6.12.6-xen0  root=/dev/sda1 ro console=tty0
```

Hands-on: netwerktoegang voor de domU
Om nu een domU te maken dienen verder een aantal zaken georganiseerd te worden in de dom0. De eerste is op het gebied van netwerk. Dit gebeurt met de standaard bridging tools (zie figuur voor een overzicht). Als de machine maar een netwerkinterface heeft dan volstaat de default Xen configuratie.
Xen kan echter omgaan met meerdere NIC's. Daarvoor dienen een paar aanpassingen gedaan te worden.



De conventionele manier is om dit met de brctl commando's helemaal zelf te doen. In een geval van een machine met twee ethernetdevices kan dit er bijvoorbeeld zo uitzien:

```
# bridge 0
/usr/sbin/brctl addbr xen-br0
/usr/sbin/brctl stp xen-br0 off
/usr/sbin/brctl setfd xen-br0 0
/sbin/ip link set eth0 up promisc on
/sbin/ip link set xen-br0 up
/usr/sbin/brctl addif xen-br0 eth0

# bridge 1
/usr/sbin/brctl addbr xen-br1
/usr/sbin/brctl stp xen-br1 off
/usr/sbin/brctl setfd xen-br1 0
/sbin/ip link set eth1 up promisc on
/sbin/ip link set xen-br1 up
/usr/sbin/brctl addif xen-br1 eth1
```

En een IP-adres voor de dom0

```
/sbin/ifconfig xen-br0 172.17.1.5 netmask 255.255.255.0 up
route add default gw 172.17.1.1
```

Vanaf (in ieder geval Xen 3.0.3) kan het opzetten van networking met meerdere NIC's eenvoudiger door gebruik te maken van het Xen framework hiervoor (python). Dit gaat bijvoorbeeld door het creeren van een script met de naam `/var/xen/scripts/my-3nic-network-subscript` met de volgende inhoud:

```
#!/bin/sh
BASEPATH=/etc/xen/scripts/
"$BASEPATH/network-bridge" "$@" vifnum=0 netdev=eth0 bridge=xenbr0
"$BASEPATH/network-bridge" "$@" vifnum=1 netdev=eth1 bridge=xenbr1
"$BASEPATH/network-bridge" "$@" vifnum=2 netdev=eth2 bridge=xenbr2
```

Vervolgens dient in /etc/xen/xend-config.sxp de regel met network-script verwijderd te worden, en de onderstaande regel ervoor in de plaats toegevoegd te worden:

```
(network-script my-3nic-network-subscript)
```

Hands-on: het root-filesysteem voor domU
Nu dient diskruimte gereserveerd te worden binnen de dom0 om plaats te maken voor een domU. Een minimaal goed Linux systeem heeft een swap en een root partitie nodig. Zo ook een domU, dus deze twee partities dienen gereserveerd te worden binnen de dom0. Er zijn twee keuzes, dit kan middels loopmounted files worden gedaan of met vrije, echte partities. Deze echte partities kunnen natuurlijk ook logical-volumes zijn die op de dom0 worden gemaakt middels LVM op de dom0.

De swappartitie kan eenvoudig met het commando mkswap van de juiste inhoud worden voorzien vanuit de dom0 :-) De root partitie dient met een distributie geladen te worden. Dat kan bijvoorbeeld met de distro-tool daarvoor. Op een debian systeem kan dat middels:

```
debootstrap --arch amd64 etch /opt/xen/stock-rootfs/debian-etch
```

Een andere aanpak is om een installatie van de gewenste linux, en dat kan gerust een andere distro zijn, te doen op een ander stuk hardware (of de tweede partitie van de fysieke machine). Door nu vanaf CDrom of een andere partitie te booten kan deze distro-installatie met dd overgebracht worden naar de toebedeelde partitie op de dom0. De kernel van dit systeem wordt vanzelfsprekend niet gebruikt, en zelfs geheel /boot kan uit deze domU partitie (niet die van de dom0) verwijderd worden. Na aanpassing van de fstab file binnen dit overgebrachte filesysteem en toevoeging van /lib/modules/<xenkernel-versie> van de betreffende domU kernel, kan het systeem gebruikt worden als domU.

Hands-on: domU config
Voor het starten is er een config file nodig voor de betreffende domU. Deze dient er ongeveer zo uit te zien:

```
kernel = "/opt/xen/kernels/x86-64/domU/vmlinuz-2.6.12.6-xenU"
memory = 250
name   = "mijndomu1"
nics   = 2
vif    = [ 'mac=aa:00:00:00:00:76, bridge=xen-br0' ]
disk   = [ 'phy:mapper/vg01-domU1slash,hda1,w' , \
           'file:/var/xen/domU1/swap01.swp,hda2,w' , \
           'phy:mapper/vg01-domU1slashhome,hda3,w']
root   = "/dev/hda1 ro"
extra  = "2"
```

Indien in de dom0 de xend daemon up&running is kan met het commando:

```
xm create mijndomu1 -f /var/xen/domUs/mijndomu1.conf
```

de domU gestart worden.

Met het commando

```
xm console mijndomu1
```

wordt toegang tot de console verkregen alsof het een normale linux-machine betrof.

Hands-on: live-migration en shared-storage

Een interessante Xen feature is de live-migratie van een domU tussen twee dom0's. Voorwaarde hiervoor is echter dat beide dom0's toegang hebben tot de betreffende diskpartities waarmee de dumU opgebouwd is. Een laboratorium opstelling zou gemaakt kunnen worden door de dumU met loop-mounted files op te zetten, vanaf een NFS-shared filesysteem. Om iets meer naar een productiesituatie toe te komen is echte shared storage nodig.

De standaard basis oplossing hiervoor is een SCSI-diskcabinet met dubben aangesloten kabels. Om het nog betaalbaarder te maken is echter gebruik gemaakt van de shared-storage feature in het firewire protocol. Er is gebruik gemaakt van een maxtor onetouch-III firewire disk met interne raid-0 feature en een OxfordSemiconductor924 chip en aan de achterzijde twee fw800 connectoren. Hiermee is het eenvoudig om met behulp van twee kabels twee machines aan te sluiten.

Voor de firewire storage maakt linux gebruik van de sbp2 module. In de meeste distributies heeft deze default niet de optie voor shared gebruik enabled die het mogelijk maakt dat de shared-storage feature daadwerkelijk benut kan worden. Het betreft de module optie exclusive_login, die de waarde 0 dient te krijgen. In een debian systeem is dit bijvoorbeeld te regelen door in de configfile /etc/modprobe.d/sbp2 de regel:

```
options sbp2 exclusive_login=0 serialize_io=1
```

op te nemen. Indien de module met de initial-ramdisk al wordt geladen, dan dient deze even te worden her-gegenereerd voor het beste resultaat.

Ten behoeve van live-migratie dient als laatste in de Xen-configuratie op beide dom0's een netwerk listener voor de communicatie van de beide xen-daemons met elkaar te worden geactiveerd. Via deze weg wordt het memory van de draaiende domU vanuit de eerste dom0 naar de tweede dom0 doorgegeven.
De configuratie is gedaan door de twee opties met relocation in de naam te voorzien van de hostnaam van de “andere” dom0 node in de file:

```
/etc/xen/xend-config.sxp
```

De eerste ervaringen

Na booten van de fysieke machine met een xen-kernel is al bijna een dom0 ontstaan. Een memory listing met free geeft al een iets onverwachte waarde (dit is op een machine met 2G fysiek memory):

```
             total       used       free     shared    buffers     cached
Mem:        123464     118060       5404          0      32592      37076
-/+ buffers/cache:      48392      75072
Swap:      1959888          0    1959888
```

Voor functioneren dient echter ook de xen-daemon te worden gestart. Deze daemon is overigens in python geschreven. Starten kan dus eenvoudig door het executable-script /usr/sbin/xend aan te roepen. Bij een dom0 die gemaakt is met distrbutie geleverde software zal de xend waarschijnlijk overigens reeds gestart zijn middels de automatisch meegeinstalleerde rc-scripts. In de process listing met zijn met een grep op xen de volgende processen zichtbaar:

```
root         6  0.0  0.0     0     0 ?        S<    2005   0:00 [xenwatch]
root         7  0.0  0.0     0     0 ?        S<    2005   0:00 [xenbus]
root       703  0.0  0.0     0     0 ?        S     2005   0:13 [xenblkd]
root      2687  0.0  0.8  4028   988 ?        Ss    2005   0:06 xenstored --pid-file=/var/run/xenstore.pid
root      2689  0.0  5.9 129356 7324 ?       Ss    2005   0:05 python /usr/sbin/xend start
root      2690  0.0  1.5 15192 1908 ?        Ss    2005   0:04 xenconsoled
```

Nu kunnen met het commando 'xm info' gegevens over de fysieke machine en de xen status tevoorschijn gehaald worden:

```
system                 : Linux
host                   : mijndom0
release                : 2.6.12.6-xen
version                : #1 SMP Thu Feb 2 08:00:47 UTC 2006
machine                : x86_64
nr_cpus                : 1
nr_nodes               : 1
sockets_per_node       : 1
cores_per_socket       : 1
threads_per_core       : 1
cpu_mhz                : 2010
hw_caps                : 078bfbff:e1d3fbff:00000000:00000010
total_memory           : 2048
free_memory            : 339
xen_major              : 3
xen_minor              : 0
xen_extra              : .1
xen_caps               : xen-3.0-x86_64
platform_params        : virt_start=0xffff800000000000
xen_changeset          : unavailable
cc_compiler            : gcc version 3.3.5 (Debian 1:3.3.5-13)
cc_compile_by          : root
cc_compile_domain      : localdomain
cc_compile_date        : Thu Feb  2 00:53:20 UTC 2006
```

Een ander bruikbaar commando is 'xm list' waarmee een lijstje bovenwater komt van de verschillende virtuele machines. Het valt op dat de dom0 er zelf ook tussen staat.

```
Name                                ID Mem(MiB) VCPUs State  Time(s)
Domain-0                             0      123     1 r----- 60592.6
projadm                             70      150     1 ------  5082.9
router1                             65       65     1 -b----  1644.4
fileserver                          72      128     1 ------   127.4
web2                                74      150     1 -b----   396.6
ldap1                               46       66     1 -b----  4316.3
dns2                                47       64     1 -b----  1937.7
dns1                                48       64     1 -b----  1685.7
ldap2                               49       64     1 -b----  2675.0
mail                                51      250     1 ------ 14995.7
mysql5                              52      150     1 -b----  5653.0
mysql4                              53      128     1 ------ 16049.1
web3                                54      110     1 -b----  5874.9
jukebox                             58      128     1 ------ 36618.5
proxy                               63      100     1 -b----  1409.4
```

Een nog handiger overzichtje op een druk werkende dom0 kan met 'xm top' in beeld gezet worden. In analogie met de bekende top wordt ook dit lijstje realtime ververst, zodat een aardige indruk van de belasting van het systeem gekregen kan worden.

```
xentop - 18:49:53   Xen 3.0.1
15 domains: 1 running, 14 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown
Mem: 2096700k total, 1750508k used, 346192k free    CPUs: 1 @ 2010MHz
      NAME  STATE   CPU(sec) CPU(%)     MEM(k) MEM(%)  MAXMEM(k) MAXMEM(%) VCPUS NETS NETTX(k) NETRX(k) SSID
      web2 --b---       1407    0.1      97248    4.6     102400       4.9     1    1    49812   281093    0
     ldap2 --b---       1935    0.0      60388    2.9      65536       3.1     1    1   177989   349093    0
  Domain-0 -----r      60488    1.1     126192    6.0   no limit       n/a     1    8        0        0    0
   router1 --b---       1641    0.0      61392    2.9      66560       3.2     1    2 12430623 12505114    0
      dns2 --b---       2672    0.0      60384    2.9      65536       3.1     1    1   784951   625990    0
     proxy --b---        337    0.0     148452    7.1     153600       7.3     1    1   183727   127451    0
   projadm --b---       5069    0.2     148448    7.1     153600       7.3     1    1     6226    37640    0
   mammoet --b---      16022    0.1     125928    6.0     131072       6.3     1    1  1119977   992334    0
   jukebox --b---      36429    1.4     125932    6.0     131072       6.3     1    1 12305104 11832124    0
      dns1 --b---       1683    0.0      60388    2.9      65536       3.1     1    1   113568   206919    0
fileserver --b---         98    0.3     125924    6.0     131072       6.3     1    1  3316154   266139    0
     ldap1 --b---       4312    0.0      62432    3.0      67584       3.2     1    1   505205   617931    0
      mail --b---      14981    0.1     250852   12.0     256000      12.2     1    1  1712945  1102630    0
       db3 --b---       5647    0.0     148460    7.1     153600       7.3     1    1  9171205  3274093    0
      web1 --b---       5872    0.0     107488    5.1     112640       5.4     1    1   609511  1237850    0
```

De lijn van ervaringen

Veranderingen/verbeteringen sinds de eerste 2.0 in 2005

In de afgelopen twee jaar is hard ontwikkeld aan Xen, er is onder andere de volgende opvallende functionaliteit toegevoegd:

- een 64 bits versie is parallel aan de 32 bits versie (vanaf versie 3)
- een highspeed netwerk zolang de traffic binnen de hardware blijft (door minder tcp/ip controles)
- een dom0 kan tegenwoordig ook vanaf de tweede disk partitie werken
- een extra schedular is toegevoegd in de hypervisor
- de ACPI functies in de kernel bijten niet meer met de Xen functionaliteit (power-off etc.)
- hardware ondersteunde virtualisatie toegevoegd
- Inmiddels zit de xen software ook in de (testing)repositories van een aantal linux distro's. Dit maakt het een stuk eenvoudiger om Xen te gebruiken.

Performance en stabiliteit

Alle ervaringen in de afgelopen twee jaar geven over de Xen domU's de volgende subjectieve indruk:

- de performance is veel beter dan UML, met name opvallend vwbt. disk i/o
- uitstekend bruikbaar in de praktijk; op licht belaste systemen is in feite geen verlies van performance merkbaar wanneer deze vergeleken worden met een non-gevirtualiseerde machine.
- Met behulp van Xen geparavirtualiseerde machines blijken prima robuust (de bekende unix uptimes blijken ook met Xen mogelijk)

32 en 64 bit (op x86-64 hardware)

Vanaf het moment dat er een 64bits versie van Xen beschikbaar kwam, zijn de 32 en 64 bits versies naast elkaar bekeken en gebruikt. Dit leverde de volgende ervaringen op:

- indien gekozen wordt voor een 64 bit hypervisor en dom0 kernel, dan zullen de domU kernels ook 64 bit moeten zijn (in paravirtualisatie mode)
- in Linux is het bijna probleemloos om een 32bits userspace distributie te gebruiken icm. een 64 bit kernel. Ook op relatief kleine systemen met minder dan 4 GB memory, lijkt deze combinatie een prima keuze optie.
- dit geeft effectief het voordeel van gebruik van 32 en 64 bits userspace domU's door elkaar (lees: de root-filesystemen) op een stuk hardware.
- indien iptables of ipsec wordt gebruikt op in een 32 bits userspace met een 64 bits kernel, dan werkt dit niet. Een workaround is om de 64 bits versies van iptables en/of ipsec tools te installeren.
- 64 bit lijkt behoorlijk sneller
- 64 bit alleen als versie 3.x of hoger beschikbaar; dit is inmiddels echter nauwelijks een probleem meer.
- de bitness van een solaris-x86 domU kan momenteel alleen hetzelfde zijn als de bitness van de dom0
- er is momenteel ontwikkeling gaande om 32 en 64 bit domU kernels wel mixbaar te maken (mogelijk al vanaf xen 3.0.5)

- mixen van 32 en 64 bit kernels is vanzelfsprekend wel mogelijk bij de hardwareondersteunde virtualisatie mode (maar buiten de scope van deze paper).

Online aanpassingen aan de dumU configuratie
Aan een draaiende domU kunnen een aantal aanpassingen online worden uitgevoerd. Deze worden door middel van de xen-tool xm op de dom0 gedaan. Het betreft:

- het toevoegen van een extra disk-partitie (met fdisk in de domU is de extra partitie direct te zien)
- het toevoegen van een ethernet-device

De aanpassingen zijn niet permanent, door ze tevens toe te voegen in de start-config file worden ze bij de volgende boot ook meegenomen.

Op ongeveer dezelfde manier is online tuning van de domU mogelijk voor de parameters:

- memory (via de parameters maxmem en memory)
- vCPU's (via de parameter max cpu, alleen op een multi-cpu dom0 :-)

Keuzes voor betrouwbaarheid/productiewaardgheid:
Voor productiesystemen zijn de zogenaamde phy-devices als opslagmedium voor de partities van de verschillende domU's de aangewezen keuze. Deze zijn robuster door de afwezigheid van een tussenliggende buffercache en ze blijken in de praktijk sneller. Daarnaast is het zo dat de loopmounted filesystemen default begrenst zijn op plusminus acht, zodat voor grote testopstellingen zowieso een aanpassing ter vergroting van dit aantal gemaakt moet worden op de dom0.

Het gebruik van LVM in de dom0 om de partities flexibel te kunnen uitdelen werkt goed

Valkuilen:
Er zijn een paar issue's die bij het gebruik van Xen naar vooren kunnen komen:

- Lilo is niet te gebruiken als bootloader voor de dom0, grub is de enige keuze
- Door de highspeed networking feature die in versie 3 van Xen geintroduceerd is, kunnen er problemen op indien in de setup een firewall als domU wordt gebruikt. Het probleem ontstaat als een domU niet in de gaten heeft dat zijn IP-verkeer buiten de fysieke machine komt, maar dit wel gebeurt. Dit is het geval als de firewall twee netwerkkaarten heeft en netwerk-adres-translatie uitvoert op het verkeer van de domU. Een workaround hiervoor is om (per domU die het probleem heeft) de tcp/ip checksum offloading geheel te disablen (met het commando `ethtool -K eth0 tx off`)
- Bij gebruik van 32 bit kernels is geen native "Thread Local Storage" (TLS) mogelijk. Dit schijnt een algemeen virtualisatie issue te zijn, en dus niet geheel uniek voor Xen. De automatische emulatie mode zorgt er echter voor dat de applicaties die deze feature gebruiken wel blijven werken, maar veel langzamer. Er bestaan overigens speciale xen-vriendelijke versies van de library.

Live-migration en shared storage:
De live migration werkt zeer eenvoudig en goed. Er is in Xen geen heartbeat oid. ingebouwd, dus zonder deze toevoegingen is het vooral een beheer-feature om van dom0 te kunnen wisselen zonder downtime op de domU.
In het tweede deel van de presentatie “hands on Xen” wordt uitgebreider ingegaan op het fenomeen live-migration, en op alle zaken die daarbij voor een productie situatie belangrijk zijn.

Bij de firewire shared-storage is het opletten dat de juiste chipset in het firewire disk-cabinet aanwezig is. Niet alle chipsets ondersteunen blijkbaar het 'shared' gebruik door twee nodes, laat staan 3 of 4 (de Maxtor stopt bij twee). De OxfordSemiconductor chips lijken een minimale succesfactor te zijn voor een 2 node systeem.
Een productiewaardig diskcabinet dient echter een goede hardware disk-mirroring aan boord te hebben. Daarnaast is een goede probleem detectie en liefst een online recovery feature bij eventuele problemen met de losse harddisks nodig. Vereist is eigenlijk dat de configuratie van het cabinet vanuit het linuxsysteem zelf gedaan kan worden. De markt bied een product met deze specificaties momenteel echter nog niet aan (voor zover bekend). Hiermee lijkt de inzet van firewire als shared-storage toch enigzinds beperkt tot laboratorium opstellingen.

Beheer en management informatie:
Een goede hardware consolidatie is niet alleen technisch interessant, ook voor wat betreft accounting ten behoeve van beheer en management informatie zijn slagen te maken. Doordat de hypervisor de verschillende domU's technisch in de gaten houd, blijven ook een paar handige parameters per domU bewaard. Dit zijn:

- gebruikte cpu cycles
- netwerk traffic

Beide zijn vanuit de dom0 afleesbaar middels xm.

Conclusie

- Onder lichte belasting hebben meerdere Xen systemen probleemloos voor langere tijd gefunctioneerd, vrijwel zonder problemen. De weinige opgetreden problemen zijn zeer waarschijnlijk te wijten geweest aan slechte hardware (test-opstellingen op consumer-grade hardware).
- Test-opstelling waarop eenvoudige disk-benchmarks zijn uitgevoerd gaven eveneens prima resultaten, deze zijn niet doorgetest op de lange duur.
- Hoe Xen zich blijft gedragen als er op een multi-CPU systeem binnen een domU een database met een groot transactie volume wordt gedraaid (bijvoorbeeld voor een telco-billingdatabase of een beurstradingsysteem), is op dit moment niet bekend. Test en/of productieopstellingen zijn nog niet gemaakt. Op basis van de ervaringen is de verwachting echter dat beschreven opzet ook voor veel groter ijzer zullen werken.
- Xen lijkt op basis van de nu bekende gegevens een goede technologie voor hardware-consolidatie van x86 omgevingen. Met de groei van de capaciteit van de hardware in dit segment, lijkt een oplossing in zicht te komen die zowel prijs technisch als it-architectonisch verantwoord is.

Resources:

- http://www.cl.cam.ac.uk/Research/SRG/netos/xen/
- http://www.xensource.com/xen
- mailinglist: xen-users@lists.xensource.com
- http://docs.solstice.nl/ (in de wiki)

Over de auteur:

Luc Nieland (luc@nieland.net)
Sinds aan het eind van zijn studie een aantal indigo blauwe kasten een vastgelopen moleculair biologisch onderzoek weer richting konden geven middels een min of meer berekend drie dimensionaal structuur model van een eiwit, zijn Unix systemen niet meer uit Luc's werk verdwenen. De afgelopen tien jaar heeft hij zich gespecialiseerd in de automatisering zelf. Hij was bij verschillende organisaties betrokken bij beheer, bouw en migratie van systemen en databases ten ten behoeve van bedrijfs kritische applicaties. Momenteel is hij als Unix infrastructuur- en migratiespecialist werkzaam bij BI expertisehuis Centennium.