

Computersystemen zijn meer en meer een onmisbare schakel geworden in allerlei processen uit het dagelijks leven. De vraag naar 'highly available' systemen neemt toe. En high availability begint een hype te worden. Erik Meinders legt uit wat high availability is, welke alternatieven er zijn en waarom de jacht op de 'vijf negens' lang niet altijd zinnig is.

High availability heeft dosis nuchterheid nodig

Random hoge beschikbaarheid leven veel verkeerde verwachtingen

De samenleving is sterk afhankelijk geworden van computersystemen, en dat zal alleen nog maar verder toenemen onder invloed van de '24-uurs economie' en de globalisering (denk aan e-commerce). En daarmee groeit het belang van de beschikbaarheid van die systemen. We zien de vraag naar systemen met een hoge graad van beschikbaarheid (high availability) de laatste jaren dan ook toenemen. Met name in omgevingen waar mainframes zijn vervangen door mini's leeft sterk de indruk dat met de introductie van die laatste de kosten wellicht wat zijn gereduceerd, maar de beschikbaarheid ook niet meer is wat het geweest is. Leveranciers van hard- en software spelen daarop in door systemen te ontwikkelen die 'verhoogd' beschikbaar zijn. Er zijn echter op dit gebied nogal wat onduidelijkheden en helaas vaak ook teleurstellingen. Het koesteren van realistische verwachtingen over wat high availability wel en niet kan betekenen voorkomt teleurstellingen en maakt het mogelijk de 'beloften' van de diverse leveranciers in het juiste licht te bezien.

High availability en fault tolerance

High availability (HA) wordt in het Nederlands ook wel hoge of 'verhoogde beschikbaarheid' genoemd. Het laatste geeft waarschijnlijk

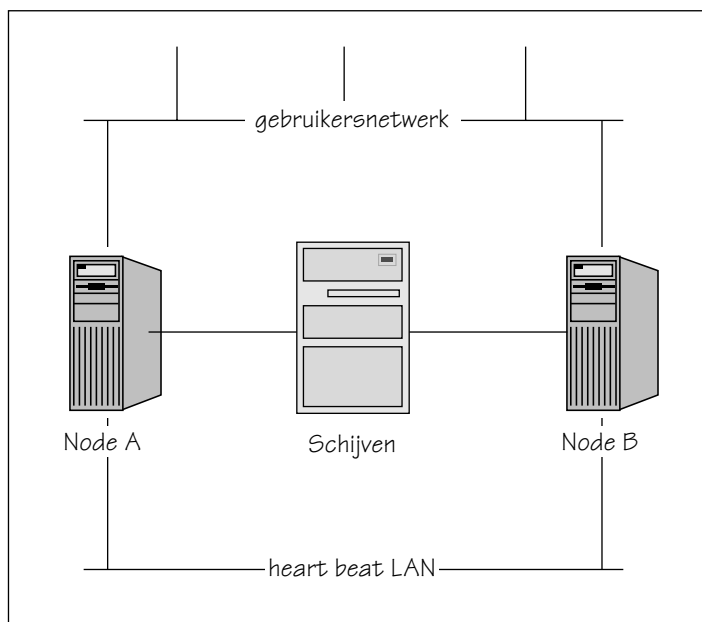
het beste aan waar het hier om gaat. De beschikbaarheid van 'het systeem' wordt opgevoerd door zodanige maatregelen te treffen, dat bij uitval van een van de componenten de beschikbaarheid van het systeem als geheel niet in gevaar komt.

High availability en *fault tolerance* zijn twee verschillende zaken die dikwijls met elkaar worden verward. Laten we proberen dit misverstand weg te nemen.

HA vaak in clusters

HA-systemen worden vaak opgebouwd door componenten redundant uit te voeren die op zichzelf niet zijn ontworpen met verhoogde beschikbaarheid als uitgangspunt. Een voorbeeld hiervan is *disk mirroring*. Twee standaard schijven worden -al dan niet softwarematig- gelijk gehouden, zodat bij uitval van (bijvoorbeeld de voeding van) een van de schijven de andere schijf het computersysteem van de data kan blijven voorzien. Voor de gebruiker wordt de uitval van de schijf dus gemaskeerd.

HA-computersystemen bestaan vaak uit zogenaamde clusters, waarbij meerdere systemen elkaars werkzaamheden in geval van calamiteiten kunnen overnemen. Daarbij worden alle belangrijke componenten zodanig redundant uitgevoerd, dat er niet één component is aan te wijzen waarvan de uitval zal leiden tot het niet langer be-



Afbeelding 1: Een eenvoudig cluster.

schikbaar zijn van het systeem: een *single point of failure* (SPOF) wordt vermeden.

Fout-tolerant: in design

Bij een fault tolerant-systeem (FT-systeem) is in het ontwerp de beschikbaarheid ervan als uitgangspunt genomen. Alle componenten zijn te vervangen zonder dat het systeem daarvoor uitgeschakeld moet worden. Ook dit wordt bereikt door componenten redundant uit te voeren, maar dan tijdens het design van de machine, in plaats van achteraf bij het ontwerpen van een cluster.

Toenemende complexiteit van de cluster-omgeving en schaarste op de arbeidsmarkt leiden tot 'instortingsgevaar' bij het beheer

De leverancier van dergelijke systemen zal doorgaans garanties kunnen geven over de beschikbaarheid van een systeem. FT-systemen zijn over het algemeen duurder dan HA-clusters met een zelfde capaciteit, al is het verschil de laatste jaren drastisch verkleind. Wel geldt dat clusters op dit moment nog steeds beter schalen in grootte. Anders gezegd: een FT-oplossing voor heel zware toepassingen is niet altijd beschikbaar.

Beheerlast

Anderzijds voegt een cluster in vergelijking tot een FT-systeem wel extra complexiteit toe aan het beheer. Een FT-systeem is doorgaans zo opgezet, dat louter de eigenschap FT de beheerder niet met veel extra beheerlast opzadelt. Daarentegen vereist het beheer van een cluster specialistische kennis en heeft het cluster ook invloed op beheerpro-

cedures en -processen. In alle beheertaken dient men zich er steeds van te verwittigen dat de te nemen stap geen invloed heeft op de beschikbaarheid, door bijvoorbeeld een SPOF te introduceren. In een HA-omgeving moeten veelal zaken op alle nodes in het cluster worden uitgevoerd (denk aan het configureren van een printer), hetgeen extra werk voor de beheerder inhoudt en de kans op fouten en vergissingen vergroot. Daar komt bij dat een dergelijke fout of vergissing vaak pas aan het daglicht komt op het moment dat er zich een calamiteit voordoet - doorgaans niet het meest geschikte moment om met extra problemen te worden geconfronteerd.

Belofte en werkelijkheid

Leveranciers overtreffen elkaar in beloften aangaande beschikbaarheid. Vijf negens, ofwel 99,999% beschikbaarheid, lijkt binnen handbereik. Dat houdt in dat een systeem zo'n vijf minuten per jaar niet beschikbaar is. Bij fault tolerant-systemen wordt zelfs hardop gesproken over 100% beschikbaarheid.

Beschikbaarheid is een begrip dat voor iedereen iets anders betekent. Jaren geleden was ik betrokken bij het beheer van tientallen Unix-systemen bij een klant waar de gevleugelde uitspraak 'het licht brandt' de ronde deed. De plaatselijke beheerder vond namelijk dat het systeem

Cluster-technologieën

Een eenvoudig cluster kan bestaan uit twee systemen, die beide toegang hebben tot hetzelfde disk-subsysteem, zoals in afbeelding 1. Er zijn verschillende manieren waarop de systemen verhoogd beschikbaar kunnen worden gemaakt. Een oplossing is dat een van de systemen actief is en het andere stand-by totdat het actieve systeem niet langer in staat is de werkzaamheden uit te voeren. Dan start het stand-by systeem op (bijvoorbeeld van de boordisk van het niet-langer-actieve systeem). Het behoeft geen betoog dat het stand-by systeem niet voor andere taken dan stand-by gebruikt kan worden; het kan immers elk moment de taak van de productiemachine moeten overnemen, en dat vereist een reboot. Dit principe wordt *cold stand-by* genoemd.

Een andere aanpak is die waarin de machines tegelijk actief zijn; veelal één systeem als productieserver, de andere als test/ontwikkel-server. Valt de productieserver uit, dan zal de andere server de productieprocessen overnemen door deze opnieuw op te starten op de test/ontwikkel-server. Eventueel worden de test/ontwikkel-processen daarbij gestopt of verlaagd in prioriteit. Het wezenlijke verschil met cold stand-by is dat men in dit geval beide systemen zinvol kan inzetten als er geen calamiteit is. Bovendien is de tijd dat de applicatie niet beschikbaar is korter, omdat het failover-systeem niet opnieuw hoeft te booten. Producten als MC/ServiceGuard (HP) en Sun Cluster (Sun Microsystems) vallen in deze categorie.

Een derde mogelijke aanpak is een cluster waarbij op beide systemen tegelijk dezelfde applicatie draait. Gebruikers worden verdeeld over de servers en uitval van een van de systemen zal ertoe leiden dat de gebruikers die aan dat systeem waren verbonden worden overgezet naar het andere systeem. Een voordeel van deze opzet is dat op deze manier het cluster niet alleen gebruikt wordt ter verbetering van de beschikbaarheid, maar direct ook help bij het verbeteren van performance (schaalbaarheid). Oracle heeft een toepassing genaamd Oracle Parallel Server (OPS) die dit mechanisme implementeert.

De hardware voor bovenstaande voorbeelden is grotendeels identiek. De verschillen in de oplossingen liggen voornamelijk in de gebruikte (cluster) software.

beschikbaar (want *aan*) was wanneer het power-controlelampje op het systeem brandde. Dat gebruikers er vervolgens geen gebruik van konden maken was niet zijn zorg; immers, de database was de verantwoordelijkheid van de dba, en mocht er een netwerkprobleem zijn, dan was daar de afdeling netwerkbeheer. Zo beredeneerd is een hoge beschikbaarheid eenvoudig haalbaar.

Voor een eindgebruiker ligt dat toch anders. Deze beoordeelt de beschikbaarheid op basis van het kunnen werken met het systeem. Ongeacht wát er aan schort, 'het systeem' is niet beschikbaar als de gebruiker zijn of haar werk niet kan uitvoeren.

Definiëren

Menig dispuut over beschikbaarheidspercentages komt voort uit het niet nauwkeurig definiëren hoe deze percentages dienen te worden vastgesteld. De percentages die de leveranciers van systemen ons voorschotelen gaan 'slechts' de door hen geleverde machine of cluster van machines aan. Uiteraard is dit een wezenlijk onderdeel van het geheel, maar voordat ook de gebruiker vindt dat het systeem beschikbaar is, zal toch de database moeten draaien, de netwerkinfrastructuur

Bij fault tolerant-systemen wordt zelfs hardop gesproken over 100% beschikbaarheid

correct moeten functioneren en ook de werkplek van de gebruiker *up and running* moeten zijn. De vraag is dan ook altijd hoever je moet gaan met het redundant uitvoeren van zaken. Ook hier geldt dat het geheel zo sterk is als de zwakste schakel.

Zo was ik ooit betrokken bij de bouw van een enorm SAP-cluster in Koeweit, waar de schier oneindige hoeveelheid geld die voor het project beschikbaar was ervan afstraalde. Alles was super-de-luxe en uiteraard redundant uitgevoerd; viervoudige FDDI-ringen, redundant uitgevoerde EMCs (HA-storage-subsystemen), twee tape-robots met ieder vijf tape-units per systeem en ga zo maar door. De hardware werd zelfs in twee vliegtuigen ingevlogen. Er moest echt een behoorlijke calamiteit optreden om de centrale SAP-server onderuit te laten gaan. Maar het geheel kwam terecht in één computerzaal met onvoldoende airconditioning en het kantoor netwerk waarop dit megacluster moest worden aangesloten was één enkelvoudig 10 Mb Ethernet-segment. Bij wijze van spreken was één kapotte hub voldoende geweest om het gehele kantoor netwerk -en daarmee de *toegang* tot het HA-systeem- onbruikbaar te maken. De miljoeneninvestering die daar werd gedaan voor het 'HA krijgen' van een belangrijk SAP-cluster werd voor een groot deel teniet gedaan door de omgeving waarin deze werd geplaatst.

Beheer wezenlijk anders

Een andere uitdaging in een HA-omgeving is het beheer. Wij zien dikwijls dat HA-clusters binnen enkele weken tot maanden na oplevering niet meer (geheel) functioneren. Het beheer van een HA-cluster is namelijk wezenlijk anders dan dat van een individueel systeem. Toenemende complexiteit van de omgeving en schaarste op de arbeidsmarkt leiden tot 'instortingsgevaar'. Beheerders met gedegen

Round Robin DNS

Om het IP-adres van een URL te achterhalen zoekt een browser contact met een Domain Name Server. Dit kadertje is te klein om DNS in al z'n schoonheid toe te lichten, maar het komt erop neer dat elk inter-netdomein een of meer DNS servers heeft. De DNS voor een gegeven domein kan alle hostname naar IP-adresvertalingen en vice versa voor dat gegeven domein voor zijn rekening nemen.

Round Robin DNS zorgt ervoor dat voor één hostname verschillende IP-adressen worden teruggegeven door DNS. In plaats van één systeem staan er meerdere systemen met allemaal een verschillend adres, maar met dezelfde naam. Een aanvraag aan de DNS voor een gegeven naam zal steeds met een ander adres worden beantwoord. Dit komt zowel beschikbaarheid als schaalbaarheid ten goede. Wel is het zo dat ook de achterliggende applicatie zich voor een dergelijke oplossing moet lenen.

kennis over de specifieke problemen en uitdagingen in een HA-omgeving zijn dun gezaaid. De training van beheerders wordt vreemd genoeg vaak als sluitpost op de begroting van HA-projecten gezien of wordt op een zodanig laat tijdstip gepland, dat het cluster inmiddels onbedoeld is afgebroken. Er hebben dan ook al projecten bestaan met als doelstelling niet het opbouwen, maar juist het ontmantelen van een cluster. De toegevoegde complexiteit van het beheer werd als een minstens zo groot knelpunt gezien als het eventueel uitvallen van een van de systemen. Ook hernieuwd nadenken over wat nu echt belangrijk is speelt bij dergelijke projecten vaak een rol.

De hype

Nu HA een hype aan het worden is, wil "iedereen" HA-systemen. Maar, pas op, HA mag dan het antwoord zijn, laten we de vraag niet uit het oog verliezen. Natuurlijk, de ene oplossing is nog mooier dan de andere, en *feature-fighting* door leveranciers is een veel voorkomend fenomeen in deze markt. Maar laten we de *requirements* niet uit het oog verliezen. Met welk doel wordt een HA-omgeving aangeschaft? En zijn de eisen van de opdrachtgever wel altijd even reëel? Is het noodzakelijk binnen één minuut verder te werken als het hoofdkantoor in vlammen opgaat?

HA mag dan het antwoord zijn, laten we de vraag niet uit het oog verliezen!

Technisch is erg veel mogelijk. En de 'business' verlangt ook vaak extreem hoge beschikbaarheid. De kosten die gemoeid gaan met de extra investeringen in hard- en software en training van de beheerders zijn echter navenant hoog. In een recent uitgevoerde HA-audit kwam ter sprake dat de business heel hoge eisen aan de beschikbaarheid stelde, terwijl de organisatie zodanig was ingericht, dat investeringen in IT ten laste kwamen van de afdeling automatisering. Het is zaak dat voor de interne klanten van deze automatiseringsafdeling inzichtelijk is dat deze verhoogde beschikbaarheid ook hogere kosten met zich meebrengt. Zeker als verschillende afdelingen binnen één organisatie

verschillende eisen stellen aan de beschikbaarheid van hun systemen. Technisch gesproken had men z'n zaakjes op orde; maar om in de toekomst beter in te kunnen spelen op de wensen en eisen van de (inter-

Is het noodzakelijk binnen één minuut verder te werken als het hoofdkantoor in vlammen opgaat?

ne) klant is het in zo'n geval nodig het hele systeem van budgetteren te veranderen. De klant moet extra betalen voor extra beschikbaarheid.

Alternatieven

Gezien de complexiteit van op geclusterde systemen gebaseerde omgevingen lijkt het verstandig te kijken naar alternatieve oplossingen. De FT-systemen bijvoorbeeld komen qua prijs en schaalbaarheid het HA-cluster steeds meer langszij. Zeker als de extra complexiteit van een cluster en de bijkomende problemen in de beheeromgeving worden betrokken in de vergelijking.

Zo bouwt Stratus systemen met processors en memory redundant uitgevoerd in een systeem. Voor de beheerder is deze oplossing identiek aan een 'gewoon' systeem, met als verschil dat bij het uitvallen van een

component het systeem doordraait en de component zonder downtime is te vervangen. Deze systemen zijn bijvoorbeeld beschikbaar met PA-RISC-processors, zodat daarop HP-UX kan draaien, maar binnenkort ook als 'gewone' pc's met daarop Windows 2000.

Soms kan het nog simpeler. Denk aan een (statische) website. Deze kan ook HA worden gemaakt door gebruik te maken van zaken als Round-Robin DNS (zie kader) in plaats van de diverse, vaak ingewikkelde, HA clusters.

Conclusie

Er wordt veel gesproken over beschikbaarheid en het voortdurend beschikbaar zijn van systemen acht men van toenemend belang. SLAs worden afgesproken en automatiseringsafdelingen worden beoordeeld en afgerekend op basis van de beschikbaarheid van systemen. Veelal grijpt men om een hogere beschikbaarheid te verkrijgen direct naar het middel van clustering. Dit is dan ook voor veel HA-toepassingen de meest geëigende oplossing. Als de scope van een HA-project juist is, en ook zaken als beheer en organisatie onder de loep worden genomen, kan de inzet van een HA-cluster een verbetering van niet alleen beschikbaarheid, maar ook schaalbaarheid en beheerbaarheid met zich mee brengen.

Feature-fighting door leveranciers komt veel voor in deze markt

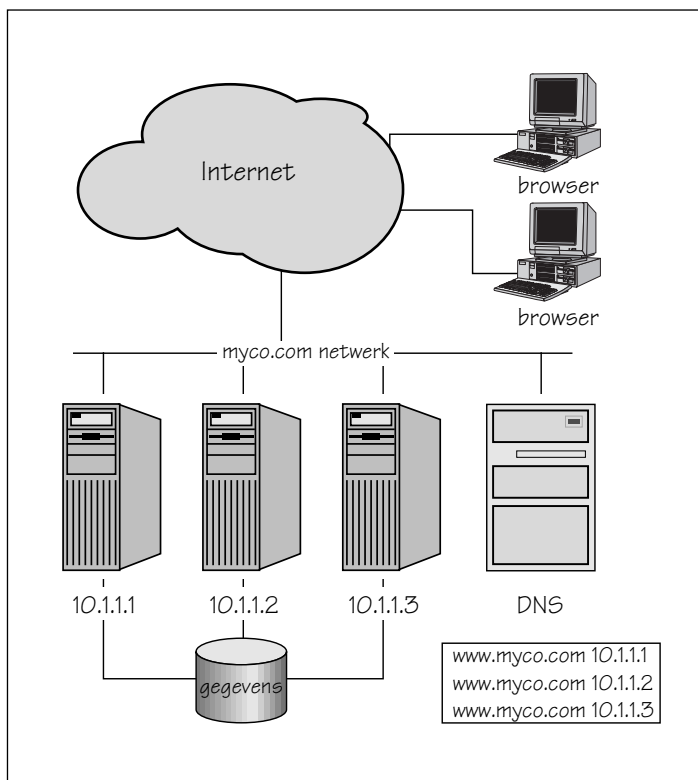
Maar het is goed om ook te kijken naar alternatieven die wellicht voor bepaalde organisaties al een voldoende verhoging van de beschikbaarheid opleveren. De alternatieven hoeven lang niet altijd technisch van aard te zijn. Een verbetering van het servicecontract met de leverancier door het aanscherpen van responstijd kan al een stap in de goede richting zijn. Belangrijk is in elk geval eerst te kijken naar de daadwerkelijk vereiste beschikbaarheid alvorens te streven naar het technisch hoogst haalbare.

Er wordt over het op zichzelf zinnige streven naar verhoogde beschikbaarheid veel zin en onzin verkondigd, en men schernt met getallen die soms gespeend zijn van elk verband met de praktijk van alledag. Er is, kortom, nog een hoop te doen ter verbetering van beschikbaarheid. Al was het maar aan een eenduidige definitie ervan.

Erik Meinders

Erik Meinders is als senior Unix-consultant werkzaam bij Open Solution Providers. Hij heeft diverse HA-clusters ontworpen en geïmplementeerd in binnen- en buitenland. Daarnaast verzorgt hij geregeld cursussen, onder meer op het gebied van high availability.

E-mail: erik@osp.nl



Afbeelding 2: Round-Robin DNS, een alternatieve aanpak die soms werkt om high availability te bewerkstelligen.